Linear Spaces of Meanings: Compositional Structures in Vision-Language Models

Matthew Trager Pramuditha Perera Luca Zancato Alessandro Achille

Parminder Bhatia Stefano Soatto

AWS AI Labs

{mttrager,pramudi,aachille,parmib,soattos}@amazon.com
zancato@amazon.it

Abstract

We investigate compositional structures in data embeddings from pre-trained vision-language models (VLMs). Traditionally, compositionality has been associated with algebraic operations on embeddings of words from a pre-existing vocabulary. In contrast, we seek to approximate representations from an encoder as combinations of a smaller set of vectors in the embedding space. These vectors can be seen as "ideal words" for generating concepts directly within embedding space of the model. We first present a framework for understanding compositional structures from a geometric perspective. We then explain what these compositional structures entail probabilistically in the case of VLM embeddings, providing intuitions for why they arise in practice. Finally, we empirically explore these structures in CLIP's embeddings and we evaluate their usefulness for solving different vision-language tasks such as classification, debiasing, and retrieval. Our results show that simple linear algebraic operations on embedding vectors can be used as compositional and interpretable methods for regulating the behavior of VLMs.

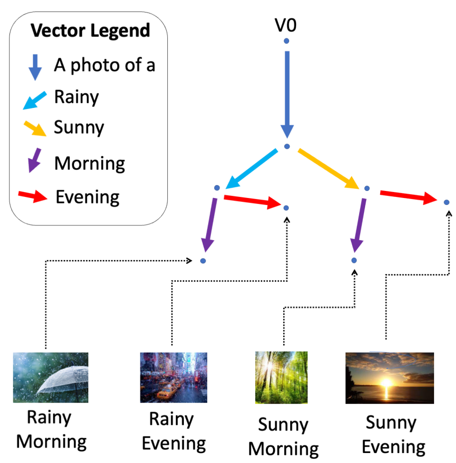

Figure 1: Words and concepts in natural language can be composed to generate complex meanings efficiently. Embeddings from transformer-based models a priori do not have a similar structure. In this paper, we argue that representations of composite concepts admit a linear decomposition based on embedding vectors that can be viewed as "ideal words."

1. Introduction

In natural language, few primitive concepts or words can be used compositionally to generate a large number of complex meanings. For example, Figure 1 shows a simple example of composed phrases {rainy, sunny} × {morning, evening}, to which one could add more factors in the form of adjectives or attributes. The hidden representations provided by a neural model, on the other hand, a priori *do not* have a similar compositional structure. In contextual text embeddings, in particular, the representation of a string of text is jointly affected by all of its tokens simultaneously, which means that there is no simple relationship between the representations of the entire text and the words that appear in it.

In this paper, we investigate the existence of latent compositional structures in the embedding space. That is, we aim to decompose composite concepts as linear combinations of embedding vectors associated with different factors, as illustrated in Figure 1. If such vectors exist, they can be treated as *ideal words* for composing new concepts directly within the representation space of the model. The first application that we envision is for vision-language models (*e.g.*, CLIP [41]) where embeddings of text labels are often used for image classification or retrieval. In this setting, linear compositionality would imply that we could classify an image with $n_1 \dots n_k$ composite labels—where n_i indicates the number of options for each factor—by comparing each image with only $n_1 + \dots + n_k$ ideal words, since by linearity the inner product of an image with a composed label is the sum of the product with the corresponding ideal words. Moreover, linear decompositions can be used for "post-hoc" manipulations of pre-trained data representations (*e.g.*, amplifying or reducing the importance of certain factors), which can be helpful to control the behavior of neural models.

In general, the meaning of words in language is always *contextual*, in the sense that their interpretation depends on any text that surrounds them. However, language would be completely impractical if words did not also have some stability in their meaning. The main benefit of the usage of words is, in fact, that meaning can be mostly inferred compositionally by combining meanings of words or phrases. There is, therefore, *a natural tension between compositionality and contextuality*: the former requires some amount of independence from context, while the latter allows for general dependencies. In a sense, our goal in this work is to consider representations of meanings that were originally learned as contextual, and to later approximate them as needed with compositional ones based on ideal words. This combines the flexibility and expressiveness of contextuality with the structural efficiency of compositionality. Our main contributions can be summarized as follows:

- We describe compositional linear structures from a geometric perspective and explain how these structures can be approximately recovered from arbitrary collections of vectors associated with a product of "factors." We also relate these structures with previous definitions of disentangled representations that were based on mathematical representation theory [26] (Section 3).
- We consider embeddings arising from visual-language models (VLMs) and show that the existence of linearly factored embeddings is equivalent to the conditional independence of the factors for the probability defined by the model. We also discuss some relaxations of this result that illustrate how linear structures may emerge even when if true data distribution satisfies weaker "disentanglement" conditions (Section 4).
- We empirically show that embeddings of composite concepts can often be well-approximated as linear compositional structures, and that this leads to simple but effective strategies for solving classification and retrieval problems in a compositional setting. We also visualize manipulations of factored embeddings using a CLIP-guided diffusion model (Stable Diffusion [42]).

2. Related Work

Compositionality has long been recognized to be a fundamental principle in cognition [20]. It has been a central in theme in Gestalt psychology [16], cognitive sciences [19], and pattern theory [24]. The main benefit of compositional representations is that they avoid the combinatorial explosion that occurs if all composed concepts are considered to be completely distinct. This property is of course a characteristic feature of natural languages, which use a fixed vocabulary for all represention, making "infinite use of finite means" (von Humboldt) [10]. However, while there is large body of work in NLP devoted to learning compositional representations of language (*e.g.*,[37, 12, 5, 22, 13]), modern text representations based on transformer architectures [47] are a priori *not* compositional in any way. Some works have studied whether compositionality is implicitly present in neural networks, for example by evaluating the ability of these models to generalize beyond the training data [27]. More relevant to our purposes, [3] proposed a framework for evaluating the compositionality of a network's internal representations, by searching for representational primitives; however, finding such compositional primitives requires solving an optimization problem. In a broad sense, compositionality can be seen as a particular way of exploiting or imposing *structure* in the inner representations of a network. It has also been argued that data representations should be concentrated in low-dimensional linear spaces [34, 9], or even be "disentangled" with respect to factors of variation in the data [26, 8, 1]. Our perspective on compositional representations is closely related to the definition of disentanglement given in [26]. As argued above, compositionality of text representations is naturally in tension with *contextuality*. Since their introduction in NLP around 2018 [40, 15], contextual text embeddings have been extremely successful, and are part of modern transformer-based architectures. The amount of contextuality in these word embeddings has been quantified using different metrics in [17].

Linear compositionality for embeddings is often associated with popular "vector analogies" that are known to roughly hold for (non-contextual) word embeddings such as word2vec [36] and GloVe [39]. Several works have proposed theoretical justifications for this property [29, 4, 25, 2, 18, 45]. To our knowledge, however, similar properties for contextual embeddings of language models have not been considered, although [46] has evaluated the performance of transformer-based models on analogy tasks. Various limitations of linear analogies have also been pointed out [31, 7].

In the context of image generation, compositional approaches for controlling the output of diffusion models have been recently proposed in [32, 48]. In particular, [48] introduced a "concept agebra" that is formally similar to our factored representations; however, their notion of "concept" is based on score representations (gradient of log-probabilities), rather than on embedding vectors, which leads to a different probabilistic characterization of compositionality. Finally, [11] introduced a method for removing biases and spurious correlations from pre-trained VLM embeddings for both discriminative and generative tasks; since their proposed approach consists in applying certain linear projections to textual embeddings (with some calibration adjustments), it can be seen as conceptually similar to an application of our ideal word decompositions.

3. Linearly Factored Embeddings

We begin by discussing from a purely geometric perspective what we mean by "linear compositionality." We consider a finite set $\mathcal{Z} = \mathcal{Z}_1 \times \ldots \times \mathcal{Z}_k$ that we view as representing a factored set of "concepts." For example, the set $\mathcal{Z}$ may be a collection of strings of text organized in a structured way, *e.g.*, according to attribute-object-context. We often write elements of $\mathcal{Z}$ as $z = (z_1, \ldots, z_k)$ with $z_i \in \mathcal{Z}_i$ and refer to z_i as the components of z. We now consider an arbitrary embedding map $r: \mathcal{Z} \to V$ of $\mathcal{Z}$ into a vector space V.

Definition 1 (Linearly factored embeddings)**.** A collection of vectors $r(\mathcal{Z}) = \{\boldsymbol{u}_z : z \in \mathcal{Z}\} \subset V$ parameterized by $\mathcal{Z} = \mathcal{Z}_1 \times \ldots \times \mathcal{Z}_k$ is *linearly factored* if there exist vectors $\boldsymbol{u}_{z_i} \in V$ for all $z_i \in \mathcal{Z}_i$ $(i = 1, \ldots, k)$ such that

$$\boldsymbol{u}_z = \boldsymbol{u}_{z_1} + \ldots + \boldsymbol{u}_{z_k}, \tag{1}$$

for all $z = (z_1, \ldots, z_k)$.

This notion is very intuitive and can be seen as a generalization of the additive compositionality that has been considered for (pairwise) analogies and word embeddings [36].

Lemma 2. *1) A collection of vectors $r(\mathcal{Z})$ is linearly factored if and only if the vector difference $\boldsymbol{u}_z - \boldsymbol{u}_{z'}$ does not depend on the components that $z, z' \in \mathcal{Z}$ share in common. 2) If $|\mathcal{Z}_i| = n_i$, then the dimension of $Span(r(\mathcal{Z}))$ is at most $1 + \sum_{i=1}^{k}(n_i - 1)$.*

It is easy to realize that if a collection of vectors $r(\mathcal{Z})$ is linearly factored, then the vectors appearing on the right of equation 1 are *never* uniquely determined. In particular, even though each $\boldsymbol{u}_{z_i}$ is associated with a value of a factor $z_i \in \mathcal{Z}_i$, that vector cannot carry any "semantic" content. However, we can recover uniqueness in the components by simply turning to a "centered" decomposition.

Lemma 3 (Centered decomposition)**.** *If a collection of vectors $r(\mathcal{Z})$ is linearly factored, then there exist unique vectors $\boldsymbol{u}_0 \in V$ and $\boldsymbol{u}_{z_i} \in V$ for all $z_i \in \mathcal{Z}_i$ $(i = 1, \ldots, k)$ such that $\sum_{z_i \in \mathcal{Z}_i} \boldsymbol{u}_{z_i} = 0$ for all i and*

$$\boldsymbol{u}_z = \boldsymbol{u}_0 + \boldsymbol{u}_{z_1} + \ldots + \boldsymbol{u}_{z_k}, \tag{2}$$

for all $z = (z_1, \ldots, z_k)$.

In the previous decomposition, the vectors $\boldsymbol{u}_{z_i}$ are now uniquely associated with the value of a factor $z_i \in \mathcal{Z}_i$, but are *relative* to the other values in $\mathcal{Z}_i$ (since they sum to zero). Similarly, the vector spaces $V_{\mathcal{Z}_i} := Span(\boldsymbol{u}_{z_i} : z_i \in \mathcal{Z}_i)$ are uniquely associated with each factor $\mathcal{Z}_i$. In our applications, we will refer to $\boldsymbol{u}_i$ as the *ideal words* of the linear factorization and to each $V_{\mathcal{Z}_i}$ as the *semantic space* associated with $\mathcal{Z}_i$. Despite its simplicity, we believe that the decomposition in Lemma 3 paints an interesting intuitive picture of linear models of "meaning." In this setting, the origin is not a universally meaningful point; for example, the origin of text embeddings does not correspond to the null string. Thus, meanings might be best viewed as an *affine space*, where the origin is only chosen as a particular reference that may depend on context. Ideal words, on the other hand, provide *relative meanings* with respect to the context.

From Lemma 2, it follows that factored representations must be very low-dimensional and, in particular, "generic" embeddings will *not* be factored. However, it is very easy to recover the nearest factored approximation for any given set of vectors $\boldsymbol{u}_z, z \in \mathcal{Z}$.

Proposition 4. *Let α_{z_i} $z_i \in \mathcal{Z}_i$ be arbitrary positive weights such that $\sum_{z_i \in \mathcal{Z}_i} \alpha_{z_i} = 1$, and define $\beta_z := \prod_i \alpha_{z_i}$ for all $z = (z_1, \ldots, z_k)$. Then, for any norm $\|\cdot\|$ induced by an inner product on V, we have that*

$$\begin{aligned} &\arg\min_{\tilde{\boldsymbol{u}}_z} \sum_{z \in \mathcal{Z}} \beta_z \|\boldsymbol{u}_z - \tilde{\boldsymbol{u}}_z\|^2, \\ &\qquad s.t.\ \{\tilde{\boldsymbol{u}}_z\} \text{ is linearly factored}, \end{aligned} \tag{3}$$

is given by $\tilde{\boldsymbol{u}}_z = \boldsymbol{u}_0 + \boldsymbol{u}_{z_1} + \ldots + \boldsymbol{u}_{z_k}$ where

$$\boldsymbol{u}_0 := \sum_z \beta_z \boldsymbol{u}_z,\ \boldsymbol{u}_{z_i} := \frac{1}{\alpha_{z_i}} \sum_{\substack{z' = (z'_1, \ldots, z'_k) \\ z'_i = z_i}} \beta_z \boldsymbol{u}_{z'} - \boldsymbol{u}_0. \tag{4}$$

This fact shows that computing linearly factored approximations amounts to performing simple weighted averages of the original vectors. In many cases, we will consider $\alpha_{z_i} = \frac{1}{n_i}$ and $\beta_z = \prod \frac{1}{n_i}$, however it can be useful to allow for additional "knobs," as the following example illustrates.

Example 5. One of our main motivations to consider linearly factored structures is to approximate (pre-trained) contextual text embeddings to obtain representations that are *interpretable* and *compositional*. More concretely, assume that each factor $\mathcal{Z}_i$ represents a finite collection of strings and that the representation $r : \mathcal{Z}_1 \times \ldots \times \mathcal{Z}_k \to V$ is defined by concatenating strings and then embedding the result using a contextual language encoder. For a very simple example, consider

$$\mathcal{Z} = \{\text{a blue, a red, a green}\} \times \{\text{bike, house}\},$$

which leads to six possible strings and six distinct embedding vectors. Using Proposition 4, we can easily find a factored approximation $\boldsymbol{u}_{(col,obj)} \approx \boldsymbol{u}_0 + \boldsymbol{u}_{col} + \boldsymbol{u}_{obj}$, where $\boldsymbol{u}_{col}$ and $\boldsymbol{u}_{obj}$ are the ideal words representing a particular object and color from $\mathcal{Z}$. As we will see, these vectors can be used for semantic manipulations of embeddings. Note that ideal words are not the same as the encodings of the original words or substrings. In fact, quite intuitively, the meaning of ideal word vectors is determined entirely by the way in which the corresponding string interacts with other factors. For example, we have $\boldsymbol{u}_{\text{green}} = \alpha_{car}\boldsymbol{u}_{(\text{green car})} + \alpha_{house}\boldsymbol{u}_{(\text{green house})} - \boldsymbol{u}_0$ where $\boldsymbol{u}_0$ is the mean of all six embeddings. In this particular example, "green house" has distinct contextual meaning, but this can be controlled by using appropriate weights, if desired. See Section 5 and Figure 3 for more discussions on similar examples.

We conclude this section by pointing out a connection between linearly factored embeddings and a notion of "disentangled representations" proposed in [26]. We refer to the Appendix for a short summary of the relevant mathematical background and for additional discussions. In a broad sense, we can say that an embedding map $r : \mathcal{Z} \to V$ into a vector space V is "linearly compositional" with respect to some group of transformations G if 1) G acts on the set $\mathcal{Z}$ 2) G acts on V as invertible linear transformations, and 3) r is a G-morphism, that is, if $r(g \cdot z) = g \cdot r(z)$. In our case of interest, the set $\mathcal{Z} = \mathcal{Z}_1 \times \ldots \times \mathcal{Z}_k$ is a finite set of composite concepts (*e.g.*, {rainy, sunny} × {morning, evening}) and $G = \mathfrak{S}_{n_1} \times \ldots \times \mathfrak{S}_{n_k}$ is a product of symmetric groups that acts on $\mathcal{Z}$ by varying each component separately (*e.g.*, swapping "rainy" ↔ "sunny" and "morning" ↔ "evening," independently). Following [26], we say that the action of G on V is "linearly disentangled" if there exists a decomposition $V = V_1 \oplus \ldots \oplus V_k$ such that $g = (g_1 v_1, \ldots, g_k v_k)$ for all $v = (v_1, \ldots, v_k) \in V$ and $g = (g_1, \ldots, g_k) \in G$. Intuitively, this means that we can permute the different factors independently by acting with linear transformations on the embedding space. With these definitions in place we have that linear factorizations of embeddings are intimately related to disentangled compositional representations.

Proposition 6. *Let $r(\mathcal{Z})$ be a set of linearly factored vectors of maximal dimension. Then r is compositional for some disentangled action of $G = \mathfrak{S}_{n_1} \times \ldots \times \mathfrak{S}_{n_k}$ on V. Conversely, if r is compositional for a disentangled action of G, then the vectors $r(\mathcal{Z})$ are linearly factored.*

4. Linearly Factored Embeddings in Visual Language Models

In this section, we discuss linear factorizations from a probabilistic viewpoint in the context of vision-language models (VLMs). A priori, it may not be clear why the geometric notion of factored embeddings should be relevant in practice—for example, in the case of CLIP's normalized embeddings, it may seem that non-linear spherical geometry should come into play. In this section, however, we argue that vector factorizations have simple probabilistic intepretations, and in particular, we should expect these structures to be present in real data embeddings.

In the following, we write $\mathcal{X}$ for a set of texts and $\mathcal{Y}$ for a set of images (for simplicity, we consider a finite set of text and images, which will always be the case in practice). We consider a VLM that uses parametric encoders of texts $x \mapsto \boldsymbol{u}_x$ and of images $y \mapsto \boldsymbol{v}_y$ into $V = \mathbb{R}^d$ to model the conditional log-probabilities of x given y and y given x in a bilinear fashion:

$$p(x \,|\, y) = \frac{\exp \boldsymbol{u}_x^\top \boldsymbol{v}_y}{\sum_{x'} \exp \boldsymbol{u}_{x'}^\top \boldsymbol{v}_y}, \quad p(y \,|\, x) = \frac{\exp \boldsymbol{u}_x^\top \boldsymbol{v}_y}{\sum_{y'} \exp \boldsymbol{u}_x^\top \boldsymbol{v}_{y'}}. \tag{5}$$

For example, CLIP [41] uses both expressions in equation 5 to optimize a symmetric cross-entropy. This setup is similar to the one used in NLP for context-based embeddings [36] and also in transformer-based language modeling [47], the main difference being that in those cases only one of the two expressions in equation 5 is used (to model words based on context). Much of the discussion that follows can be applied to these cases as well, but we focus on VLMs for clarity.

For any given pair of embeddings $\boldsymbol{u}_x, \boldsymbol{u}_y$ there exists a unique probability $p(x, y)$ on $\mathcal{X} \times \mathcal{Y}$ compatible with these embeddings which satisfies

$$\log p(x, y) = \boldsymbol{u}_x^\top \boldsymbol{v}_y + c, \quad c \in \mathbb{R}. \tag{6}$$

In the following, we consider the distribution on $\mathcal{X} \times \mathcal{Y}$ expressed by a model and defined by equation 6. After the learning stage, this distribution should reflect a "true" distribution on the same space. We remark, however, that the embedding dimension d is in practice much smaller than the number of images or texts used in training, which means that we are actually imposing a *low-rank constraint* on the joint probability distribution. In NLP, this effect has been referred to as the "softmax bottleneck" [49].

We now consider a set of factors $\mathcal{Z} = \mathcal{Z}_1 \times \ldots \times \mathcal{Z}_k$ and assume that each $z \in \mathcal{Z}$ is represented by a string $x(z) \in \mathcal{X}$. Note that formally we could have associated factors with images rather than texts, however it is more natural to express discrete concepts as text. The factors can correspond to combinations of particular tokens (*e.g.*, attributes and objects) but the association with strings could potentially be more complex (*e.g.*, ("royal", "man") $\mapsto$ "king"). The VLM model now provides an embedding of $\mathcal{Z}$ via $z \mapsto \boldsymbol{u}_{x(z)}$.

Proposition 7. *In the setting described above, and assuming that $Span(\boldsymbol{v}_y, y \in \mathcal{Y}) = \mathbb{R}^d$, the embedding $z \mapsto \boldsymbol{u}_{x(z)}$ of $\mathcal{Z}$ is linearly factored in the sense of Definition 1 if and only if there exists functions $q_0, \ldots, q_k$ such that*

$$p(x(z), y) = q_0(y) q_1(z_1, y) \ldots q_k(z_k, y), \tag{7}$$

for all $z = (z_1, \ldots, z_k) \in \mathcal{Z}$ and $y \in \mathcal{Y}$.

Corollary 8. *Under the assumptions of Proposition 7, an embedding $z \mapsto \boldsymbol{u}_{x(z)}$ of $\mathcal{Z}$ is linearly factored if only if the factors z_i are conditionally independent given any image y.*

It is perhaps not surprising that the log-linear form of the model translates multiplicative decompositions into additive ones. It may be counterintuitive, however, that the conditional probabilities $p(z_i|y)$ as y varies actually depend on *all* of the ideal word vectors $\boldsymbol{u}_{z_i}$, since normalizing constants can change with y. Indeed we have that

$$p(z_i \,|\, y) = \exp(\boldsymbol{u}_{z_i}^\top \boldsymbol{v}_y) h(\mathcal{Z}_{j \neq i}, y), \tag{8}$$

where $h(\mathcal{Z}_{j \neq i}, y)$ is a function that depends on y and all vectors corresponding to $\mathcal{Z}_j$ with $j \neq i$. In this sense, the geometric perspective of factorization is simpler since it disregards this dependence as y varies.

The conditional independence from Proposition 7 may seem like a strict requirement and may not be obviously true in the real world. For this reason, we discuss some relaxed conditions and explain what they imply in terms of linearly factored structures. First, given an image $y \in \mathcal{Y}$, we say that the probability $p(x(z), y)$ is *mode-disentangled* (for the factor $\mathcal{Z}_i$) if

$$\arg\max_{z_i \in \mathcal{Z}_i} p(x(z_i, z_{-i}), y) = \arg\max_{z_i \in \mathcal{Z}_i} p(x(z_i, z'_{-i}), y), \tag{9}$$

for all $z_{-i} := (z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_k)$ and $z'_{-i} := (z'_1, \ldots, z'_{i-1}, z'_{i+1}, \ldots, z'_k)$. Intuitively, this simply means means that it is possible to determine the most likely value of the factor $\mathcal{Z}_i$ by disregarding all of the remaining factors. Similarly, we say that $p(x(z), y)$ is *order-disentangled* (for the factor $\mathcal{Z}_i$) if

$$\begin{aligned} &p(x(z_i, z_{-i}), y) \geq p(x(z'_i, z_{-i}), y) \\ &\quad \Longleftrightarrow p(x(z_i, z'_{-i}), y) \geq p(x(z'_i, z'_{-i}), y). \end{aligned} \tag{10}$$

for all z_{-i} and z'_{-i}. This now means that it is possible to *rank* the values of the factor $\mathcal{Z}_i$ by disregarding all of the remaining factors. It is easy to see that conditional independence implies order-disentanglement which in turn implies mode-disentanglement. If $|\mathcal{Z}_i| \leq 2$, then mode-disentanglement and order-disentanglement are equivalent.

Proposition 9 (Relaxed feasibility of linear factorizations)**.** *1) If $y \in \mathcal{Y}$ is such that $p(x(z), y)$ is mode-disentangled, then one can replace the embedding vectors $\boldsymbol{u}_{x(z)}$ with their linearly factored approximations $\tilde{\boldsymbol{u}}_{x(z)}$ from Proposition 4 (for any choice of weights) and obtain the same prediction for z given y; 2) If $p(x(z), y)$ is order-disentangled for all images y sampled from a distribution with full support over the unit sphere, then the vectors $\boldsymbol{u}_{x(z)}$ are necessarily linearly factored.*

The second part of this statement means that, roughly speaking, we should espect that imposing order-disentanglement for an increasing number of images would gradually lead to linearly factored embeddings.

Example 10. Let $\mathcal{Z}$ be of the form $\{o_1, o_2\} \times \{c_1, c_2\}$ (objects, contexts) and let $x(z)$ be the corresponding collection of strings (*e.g.*, $x(o_i, c_j) =$"a photo of a $[o_i]$ in $[c_j]$"). Then mode and order disentanglement are equivalent and mean that

$$\begin{aligned} &p(x(o_1, c_1)|y) > p(x(o_2, c_1)|y) \\ &\qquad \Leftrightarrow p(x(o_1, c_2)|y) > p(x(o_2, c_2)|y), \\ &p(x(o_1, c_1)|y) > p(x(o_1, c_2)|y) \\ &\qquad \Leftrightarrow p(x(o_2, c_1)|y) > p(x(o_2, c_2)|y). \end{aligned} \tag{11}$$

These are reasonable conditions on the probability $p(x(z), y)$ since it is normally possible to discriminate object and context in an image independently. If $p(x(z), y)$ and y satisfy equation 11, then the first part of Proposition 9 means that we can use two (approximate) "ideal word" vectors $\boldsymbol{u}_{o_1} = -\boldsymbol{u}_{o_2}$ and $\boldsymbol{u}_{c_1} = -\boldsymbol{u}_{c_2}$ instead of the four original vectors $\boldsymbol{u}_{x(o_i, c_j)}$ to assign the correct label to y. The second part of Proposition 9 means that if equation 11 holds for "all" images y (*i.e.*, vectors covering the unit sphere), then the original vectors $\boldsymbol{u}_{x(o_i, c_j)}$ are actually linearly factored.

5. Experiments

We now empirically investigate the presence and usefulness of linearly factored structures in real VLM embeddings. In all of our experiments, we use a pre-trained CLIP encoder [41][1]. Unless stated otherwise, we compute linearly factored approximations of embeddings using Proposition 4 with $\alpha_{z_i} = \frac{1}{n_i}$ and $\beta_z = \prod \frac{1}{n_i}$. We use different datasets that have a compositional nature: MIT-states [28]

[1]We use the HuggingFace implementation of CLIP with the publicly available checkpoint based on a ViT-L/14 vision transformer. See https://huggingface.co/openai/clip-vit-large-patch14

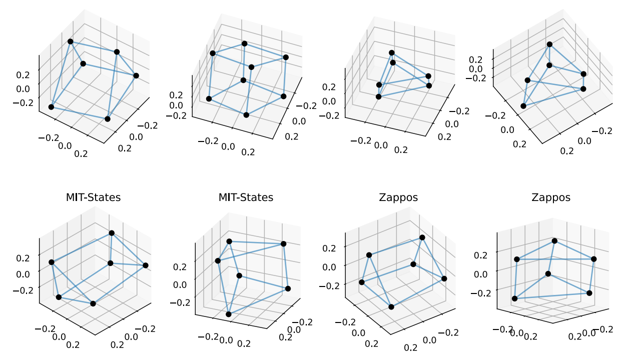

Figure 2: **Visualization of embeddings.** *Top*: projected embeddings of manually constructed strings associated with factored concepts. *Bottom:* projected embeddings for strings of the type "an image of a [a] [o]" for randomly chosen attributes and objects from MIT-states [28] and UTZappos [50]. Symmetric structures indicate that embeddings are approximately linearly factored. See text for details.

and UTZappos [50], that are image classification datasets where labels are pairs attribute–object; CelebA [33] and Waterbirds [44] in which images have a label and a spurious attribute; and DeepFashion2 [23] with PerVL annotations from [14], where the goal is to retrieve object instances from different contexts. We also include a visualization of ideal words using a CLIP-guided diffusion model (Stable Diffusion 2.1[2]) [43]. We emphasize that our goal is not to achieve state-of-the-art results, although we will see that linear manipulations can be surprisingly effective and sometimes outperform significantly more complex methods. Rather, we aim to show that linear factored structures in embedding spaces provide a useful conceptual and practical framework for *understanding* and *controlling* the behavior of pre-trained VLMs.

Visualization of embeddings. Figure 2 shows some examples of embeddings of composite strings, visualized in 3D using PCA. In the top row, we show examples of manually constructed strings. In order: "a photo of a {red, blue, pink} $\times$ {car, house}"; "a photo of a {big, small} $\times$ {cat, dog} $\times$ {eating, drinking}"; "{a photo of a, a picture of a} $\times$ {place, object, person}"; "king, queen, man, woman, boy, girl" (where one factor would correspond to male-female and the other to a generic context). In the bottom row, we present strings of the type "an image of a [a] [o]" for randomly chosen attributes and objects from MIT-states [28] and UTZappos [50] (first using two attributes and three objects, and then using three attributes and two objects). Here we always use either 2×3 or $2 \times 2 \times 2$ concepts since these factored structures have expected affine dimension 4, or linear dimension 3. The presence of roughly parallel edges and faces in these figures indicate that embeddings are approximately linearly factored. We note that in many of these examples the factorization of the concepts is already reflected in the *syntax* of the strings, *i.e.*, in the presence of repeated substrings in prompts with similar meaning. However, factorized vectors also encode semantic aspects, as can be seen in the last two examples from the first row. In the fourth example, the encoded strings have no repeated substrings, so the structure is "emergent"; in the third example, the factor corresponding to {a photo of a, a picture of a} results in an ideal word vector with a smaller norm compared to the to other directions (resulting in a "squashed" triangular prism), as one might expect since this factor is not semantically significant. We refer to the Appendix for a more in-depth discussion.

[2] https://huggingface.co/stabilityai/stable-diffusion-2-1

Compositional classification. We evaluate the usefulness of linear factored approximations for object-attribute labels of the MIT-states [28] and UTZappos [50] datasets. The default strategy for applying CLIP in a zero-shot fashion on these datasets is to use text captions such as $x(a, o)$="an image of a [a] [o]." This results in $n_{obj} \times n_{attr}$ captions that each image must be compared with. We want to explore whether the embedding vectors $\boldsymbol{u}_{x(a,o)}$ can be approximated with a linearly factored set $\tilde{\boldsymbol{u}}_{x(a,o)} = \boldsymbol{u}_0 + \boldsymbol{u}_a + \boldsymbol{u}_o$, so that inference can be performed using only $n_{obj} + n_{attr}$ embedding vectors. The intuitive choice for such vectors would be to use the representations of captions such as "image of a [a] object" and "image of a [o]." We compare this choice with using the "ideal words" associated with the original captions, where the representation of an object o is simply given by $\boldsymbol{u}_o := \frac{1}{n_{attr}} \sum_a \boldsymbol{u}_{x(a,o)}$, and similarly for attributes, as in Proposition 4 (in this setting, there is no need to remove the mean vector $\boldsymbol{u}_0$ since it is multiplied with every image vector). The resulting disjoint representations for objects and attributes ($\boldsymbol{u}_o$ and $\boldsymbol{u}_a$) are "contextualized," in the sense that they optimally approximate the original pairwise embeddings. In Table 1, "pair" refers to using the original pairwise labels, "real words" uses the embeddings of words corresponding to objects and attributes using "image of a [a] object" and "image of a [o].", while "ideal words" computes the vector ideal words for the factorization. We see that ideal words clearly outperform the *real words* baseline, and often even surpass the accuracy of *pair*. For MIT-States, using factored labels translates into using 360 vs. 28175 class vectors.

Debiasing. We can apply the decomposition into ideal words as a baseline strategy to remove contexts or biases from embeddings. The debiasing task can be formalized using the group robustness framework proposed in [44]. In this setting, we are given a collection of labels $\mathcal{Y}$ and spurious attributes $\mathcal{A}$, and we define a "group" as a pair $g \in \mathcal{Y} \times \mathcal{A}$. Assuming that each group corresponds to a

	Method	Pair Acc	Attr Acc	Obj Acc
MIT-states [28]	pair	7.7%	16.2%	47.8%
	real words	10.0%	19.3%	49.3%
	ideal words	**11.5%**	**21.4%**	**50.8%**
UT Zappos [50]	pair	**12.4%**	17.1%	**55.7%**
	real words	8.4%	10.3%	51.0%
	ideal words	10.8%	**19.2%**	55.3%

Table 1: **Zero-shot image classification results on compositional datasets.** Here "pair" refers to using all attribute-object pairs as candidate labels; "real words" refers to using labels corresponding to real words (*i.e.*, separate attribute and object labels); "ideal words" refers to using compositional labels based on ideal words. Ideal words always lead to better accuracy than real words and often even outperform pairwise labels.

probability P_g on an input space $\mathcal{X}$, the goal is to find a classifier $f : \mathcal{X} \to \mathcal{Y}$ that leads to a small gap between worst-group error and average error:

$$\max_g \mathbb{E}_{x\sim P_g}\ell(f(x), y) - \mathbb{E}_{x\sim P}\ell(f(x), y)). \qquad (12)$$

In a zero-shot setting with CLIP, classifiers are prompts that inherit biases from the dataset used in pre-training, so group robustness is not guaranteed. To address this problem, the authors of [11] propose a method for debiasing prompts that finds a projection map that makes spurious prompts irrelevant (following [6]) and then additionally regularizes the projection map to ensure that certain prompts are mapped near each other in embedding space. Here we note that a much simpler baseline would be to use ideal words to leverage the joint label-attribute representation provided by the pre-trained VL model and "average out" spurious attributes. More precisely, starting from a set of embeddings $\boldsymbol{u}_{(y,a)}$ corresponding to prompts representing each group $g = (y, a)$, ideal words suggest to define the encoding of each label y to be $\boldsymbol{u}_y := \frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} \boldsymbol{u}_{(y,a)}$. Once again, this is the same as the (shifted) ideal word corresponding to y, obtained by approximating pairwise embeddings of labels and attributes in a linearly factored way. Following [11], we evaluate group robustness of unbiased prompts on the Waterbird [44] and CelebA [33] datasets. For the Waterbird dataset, the labels are "landbird" and "waterbird," and the confounding factor is water/land background. For the CelebA dataset, the labels are "blond" and "dark" hair and the confounding factor is the binary gender. For our simple unbiasing method, we prepend prompts associated with labels with prompts associated with spurious attributes, and then average over all the spurious prompts. In both datasets, we consider exactly the same prompts for spurious attributes and labels used in [11] (see the Appendix for a description). Our results are shown in Table 2. On the CelebA dataset, our simple averaging strategy achieves a much smaller gap between average and worst group accuracy than the method proposed in [11] (1.6 vs 10.1). For Waterbird datsets, the gap is larger but comparable, and average accuracy is higher.

	Waterbird [44]			**CelebA** [33]		
	WG	Avg	Gap	WG	Avg	Gap
Zero-shot	45.3	84.4	39.1	72.8	**87.6**	14.9
Orth-Proj [11]	61.4	86.4	25.0	71.1	87.0	15.9
Orth-Cali [11]	**68.8**	84.5	**15.7**	76.1	86.2	10.1
Ideal Words	64.6	**88.0**	23.3	**83.9**	85.5	**1.6**

Table 2: **Group robustness results.** Ideal words can be used as a simple yet performant baseline for debiasing applications.

	Text Only	AvgImg+Text	PALAVRA [14]	IW
DeepFashion2 [23]	17.6 ± 0.0	21.7 ± 2.4	28.4 ± 0.7*	**37.0** ± 1.1

	IW w.o. mean removal	IW with Norm on mean	IW
DeepFashion2 [23]	22.1 ± 2.4	36.5 ± 1.4	**37.0** ± 1.1

Table 3: **Concept retrieval results.** Mean Reciprocal Rank retrieval metric on the DeepFashion2 [23] with annotations from PerVL [14]. Numbers with * are taken from [14].

Composing concepts and contexts. We perform experiments using the DeepFashion2 dataset [23] with the captions provided in PerVL [14]. This dataset contains images of 100 unique fashion items ("concepts") with textual descriptions. The task is to retrieve an image given a text query that includes a personalized concept that is specified using a small number of examples (5 samples). An example of a text query is "The [CONCEPT] is facing a glass store display." In [14], the authors propose a method called PALAVRA that trains new CLIP tokens to be associated with the custom concept; the learned tokens can then be used within natural language for retrieving images. The authors compare their method with a baseline approach dubbed "AvgIm+Text" which consists in averaging the CLIP embedding of the concept support images and of the embedded text query. This strategy is presented as the second best approach after PALAVRA. Inspired by our linear factorization of concepts and contexts, we propose to use a modification of AvgIm+Text where instead of averaging text and image embeddings, we add to the text embedding the *difference* between mean image embeddings of the specialized concept ("my shirt") and the mean embeddings of the general (coarse-grained) concept images (all images of shirts in the dataset). For a concrete example, if [CONCEPT] is a particular instance of a shirt, then the

AvgIm+Text approach would be as follows:

AvgIm+Text :

$$\begin{aligned}&\boldsymbol{u}(\text{“A person wearing [CONCEPT] sitting on a couch})\\&\approx \boldsymbol{u}(\text{“A person wearing a shirt stting on a couch})\\&\quad + \mathrm{Norm}(\mathrm{Mean}\{\boldsymbol{v}(\text{CONCEPT})\}),\end{aligned}$$

where $\boldsymbol{u}$ is the text embedding and $\boldsymbol{v}$ is the image embedding, Mean means the mean over supporting samples, and Norm means normalization. In contrast, we propose to use the following strategy:

Ideal Words :

$$\begin{aligned}&\boldsymbol{u}(\text{“A person wearing [CONCEPT] sitting on a couch})\\&\approx \boldsymbol{u}(\text{“A person wearing a shirt stting on a couch})\\&\quad - \mathrm{Mean}\{\boldsymbol{v}(\text{shirt})\} + \mathrm{Mean}\{\boldsymbol{v}(\text{CONCEPT})\}.\end{aligned}$$

Our results are shown in Table 3. Remarkably, this simple strategy that uses CLIP embeddings and *does not require any training* outperforms PALAVRA by a large margin (in our experiments, we used the implementation and evaluation code provided in [14] with only minimal changes). This modified approach can be interpreted from the perspective of linearly factored embeddings, since we are assuming that $\boldsymbol{u}(\text{context}, \text{CONCEPT}) - \boldsymbol{u}(\text{context}, \text{shirt})$ does not significantly depend on the context and can be approximated as the difference mean vectors representing the specific CONCEPT and the generic shirt. Table 3 also includes ablations for the two modifications we made w.r.t. to AvgIm+Text proposed in [14] (*i.e.* skipping the normalization step and removing the mean of the coarse-grained concept).

Visualizing ideal words. We propose to visualize the effect of linear-algebraic operations with ideal words using a CLIP-guided diffusion model (Stable Diffusion 2.1). In this setting, we compute ideal words of factored strings in the same way as before (as in Proposition 4 and Example 5), with the only difference that we now consider the encoded representation of the entire string before the final projection layer of the text encoder (treating the concatenated token representations as a long vector), since this is required for conditioning the diffusion model. An illustrative example is shown Figure 3. We mention that [48, 32] have also proposed algebraic manipulations to control visual generation in a compositional way; however both of those works perform operations on score functions rather than on embedding vectors, which means that their approach requires modifying the diffusion process. In contrast, similar to the prompt debiasing method from [11], we simply modify the prompt embeddings that condition the generation. In this paper, we use generative models as a qualitative proof of the validity of ideal words as approximations for embeddings; we leave a detailed exploration of applying these decompositions for controlling image generation to future work.

Figure 3: **Visualization of ideal words.** *First row:* images generated by Stable Diffusion with the prompt “a photo of a green house.” Because of the contextual encoder, “house” influences the meaning “green.” *Following rows:* we compute ideal words approximations for strings of the form “a photo of a [color] × [object],” using five colors and four objects. In the second row, we generate images using the vector $\boldsymbol{u}_0 + \boldsymbol{u}_{\text{green}} + \boldsymbol{u}_{\text{house}}$. Now $\boldsymbol{u}_{\text{green}}$ means green-colored because of how the string “green” composes with most objects. In the third row, we generate images using $\boldsymbol{u}_0 + \boldsymbol{u}_{[\text{color}]} + \boldsymbol{u}_{\text{house}}$ for different colors; in the fourth row, we use $\boldsymbol{u}_0 + \boldsymbol{u}_{[\text{color}]} + \boldsymbol{u}_{\text{bike}}$. The images were *not* cherry-picked or manipulated in any way. This example shows that we can generate embeddings of composite concepts by simply adding vectors in the representation space.

6. Conclusion

We have investigated compositional structures in VLM embeddings and argued that contextual text embeddings are often well-approximated by linear combinations of smaller sets of vectors. Optimal choices for these vectors are not embeddings of actual words, but rather “ideal words” that can be easily obtained as weighted averages of embeddings of longer strings of text. We showed that this simple idea can be used to design effective baseline methods for different visual language tasks (compositional classification/retrieval, debiasing, and image generation) and to control the behavior of VLMs.

In the future, we will focus on practical applications of ideal word decompositions such as compositional image generation. Furthermore, we would like to find ways of customizing ideal words using training data, for example by incorporating linear factorizations in fine-tuning strategies, or by introducing kernelized versions of these decompositions that have learnable parameters.

Finally, we remark that our discussion in Section 4 was mainly focused on embedding vectors from a single modality (text), however the strategy we used for concept retrieval in Section 5 suggests that it is possible to perform linear algebraic operations using vectors from *both* modalities (text/vision). Although it is generally known that visual and text embeddings in CLIP are not well-aligned [30], our linear manipulations actually only require for the *differences* between embedding vectors of the same modality to be aligned. Interestingly, this sort of weak alignment implies that vector representations of a concept c in any modality can be (approximately) written as

$$\boldsymbol{w}_c = \boldsymbol{w}_0 \pm \boldsymbol{w}_{\text{modality}} + \dots \tag{13}$$

where $\boldsymbol{w}_{\text{modality}}$ may be seen as the ideal word vector corresponding to the modality factor for vision/text.

References

[1] Alessandro Achille and Stefano Soatto. Emergence of Invariance and Disentanglement in Deep Representations. *arXiv:1706.01350 [cs, stat]*, June 2018. arXiv: 1706.01350. 2

[2] Carl Allen and Timothy Hospedales. Analogies Explained: Towards Understanding Word Embeddings. page 9. 2

[3] Jacob Andreas. Measuring Compositionality in Representation Learning, Apr. 2019. 2

[4] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399, 2016. 2

[5] Marco Baroni and Roberto Zamparelli. Nouns are Vectors, Adjectives are Matrices: Representing Adjective-Noun Constructions in Semantic Space. page 11. 2

[6] Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings, July 2016. arXiv:1607.06520 [cs, stat]. 7

[7] Zied Bouraoui, Shoaib Jameel, and Steven Schockaert. Relation Induction in Word Embeddings Revisited. page 11. 2

[8] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in \beta-VAE. *arXiv:1804.03599 [cs, stat]*, Apr. 2018. 2

[9] Kwan Ho Ryan Chan, Yaodong Yu, Chong You, Haozhi Qi, John Wright, and Yi Ma. ReduNet: A White-box Deep Network from the Principle of Maximizing Rate Reduction, Nov. 2021. 2

[10] Noam Chomsky. Syntactic structures. In *Syntactic Structures*. De Gruyter Mouton, 2009. 2

[11] Ching-Yao Chuang, Varun Jampani, Yuanzhen Li, Antonio Torralba, and Stefanie Jegelka. Debiasing Vision-Language Models via Biased Prompts, Jan. 2023. arXiv:2302.00070 [cs]. 3, 7, 8, 5, 6

[12] Stephen Clark. Vector Space Models of Lexical Meaning. In Shalom Lappin and Chris Fox, editors, *The Handbook of Contemporary Semantic Theory*, pages 493–522. John Wiley & Sons, Ltd, Chichester, UK, Aug. 2015. 2

[13] Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical Foundations for a Compositional Distributional Model of Meaning. page 34. 2

[14] Niv Cohen, Rinon Gal, Eli A. Meirom, Gal Chechik, and Yuval Atzmon. "This Is My Unicorn, Fluffy": Personalizing Frozen Vision-Language Representations. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, volume 13680, pages 558–577. Springer Nature Switzerland, Cham, 2022. Series Title: Lecture Notes in Computer Science. 6, 7, 8, 5

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019. arXiv: 1810.04805. 2

[16] Willis D Ellis. *A source book of Gestalt psychology*. Routledge, 2013. 2

[17] Kawin Ethayarajh. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings, Sept. 2019. 2

[18] Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. Towards Understanding Linear Word Analogies, Aug. 2019. arXiv:1810.04882 [cs]. 2

[19] Jacob Feldman. Regularity-based perceptual grouping. *Computational Intelligence*, 13(4):582–623, 1997. 2

[20] Jerry A Fodor and Ernest Lepore. *The compositionality papers*. Oxford University Press, 2002. 2

[21] William Fulton and Joe Harris. *Representation Theory*, volume 129 of *Graduate Texts in Mathematics*. Springer New York, New York, NY, 2004. 4

[22] Alona Fyshe, Leila Wehbe, Partha P. Talukdar, Brian Murphy, and Tom M. Mitchell. A Compositional and Interpretable Semantic Space. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 32–41, Denver, Colorado, 2015. Association for Computational Linguistics. 2

[23] Yuying Ge, Ruimao Zhang, Lingyun Wu, Xiaogang Wang, Xiaoou Tang, and Ping Luo. A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. *CVPR*, 2019. 6, 7, 5

[24] Stuart Geman, Daniel F Potter, and Zhiyi Chi. Composition systems. *Quarterly of Applied Mathematics*, 60(4):707–736, 2002. 2

[25] Alex Gittens, Dimitris Achlioptas, and Michael W. Mahoney. Skip-Gram - Zipf + Uniform = Vector Additivity. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 69–76, Vancouver, Canada, 2017. Association for Computational Linguistics. 2

[26] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a Definition of Disentangled Representations, Dec. 2018. 2, 4

[27] Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise?, Feb. 2020. 2

[28] Phillip Isola, Joseph J. Lim, and Edward H. Adelson. Discovering states and transformations in image collections. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1383–1391, Boston, MA, USA, June 2015. IEEE. 5, 6, 7

[29] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185, 2014. 2

[30] Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Zou. Mind the Gap: Understanding the Modality Gap in Multi-modal Contrastive Representation Learning, Oct. 2022. arXiv:2203.02053 [cs]. 9

[31] Tal Linzen. Issues in evaluating semantic spaces using word analogies. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 13–18, Berlin, Germany, 2016. Association for Computational Linguistics. 2

[32] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional Visual Generation with Composable Diffusion Models, Jan. 2023. arXiv:2206.01714 [cs]. 3, 8

[33] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild, Sept. 2015. arXiv:1411.7766 [cs]. 6, 7, 5

[34] Yi Ma, Doris Tsao, and Heung-Yeung Shum. On the Principles of Parsimony and Self-Consistency for the Emergence of Intelligence, July 2022. 2

[35] Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. Learning Graph Embeddings for Open World Compositional Zero-Shot Learning, Apr. 2022. 5

[36] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 2, 3, 4

[37] Jeff Mitchell and Mirella Lapata. Vector-based Models of Semantic Composition. page 9. 2

[38] Nihal V. Nayak, Peilin Yu, and Stephen H. Bach. Learning to Compose Soft Prompts for Compositional Zero-Shot Learning, Apr. 2022. 5

[39] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. 2

[40] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, 2018. Association for Computational Linguistics. 2

[41] 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. Learning Transferable Visual Models From Natural Language Supervision. *arXiv:2103.00020 [cs]*, Feb. 2021. 2, 4, 5

[42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022. 2

[43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models, Apr. 2022. arXiv:2112.10752 [cs]. 6

[44] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization, Apr. 2020. arXiv:1911.08731 [cs, stat]. 6, 7, 5

[45] Yeon Seonwoo, Sungjoon Park, Dongkwan Kim, and Alice Oh. Additive Compositionality of Word Vectors. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 387–396, Hong Kong, China, 2019. Association for Computational Linguistics. 2

[46] Asahi Ushio, Luis Espinosa Anke, Steven Schockaert, and Jose Camacho-Collados. BERT is to NLP what AlexNet is to CV: Can Pre-Trained Language Models Identify Analogies? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3609–3624, Online, 2021. Association for Computational Linguistics. 2

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv:1706.03762 [cs]*, Dec. 2017. arXiv: 1706.03762. 2, 4

[48] Zihao Wang, Lin Gui, Jeffrey Negrea, and Victor Veitch. Concept Algebra for Text-Controlled Vision Models, Feb. 2023. arXiv:2302.03693 [cs, stat]. 3, 8, 6

[49] Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. Breaking the Softmax Bottleneck: A High-Rank RNN Language Model, Mar. 2018. 4

[50] Aron Yu and Kristen Grauman. Fine-Grained Visual Comparisons with Local Learning. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 192–199, Columbus, OH, USA, June 2014. IEEE. 6, 7, 5