

BEYOND DECODABILITY: LINEAR FEATURE SPACES ENABLE VISUAL COMPOSITIONAL GENERALIZATION

Anonymous authors

Paper under double-blind review

ABSTRACT

While compositional generalization is fundamental to human intelligence, we still lack understanding of how neural networks combine learned representations of parts into novel wholes. We investigate whether neural networks express representations as linear sums of simpler constituent parts. Our analysis reveals that models trained from scratch often exhibit decodability, where the features can be linearly decoded to perform well, but may lack linear structure, preventing the models from generalizing zero-shot. Instead, linearity of representations only arises with high training data diversity. We prove that when representations are linear, perfect generalization to novel concept combinations is possible with minimal training data. Empirically evaluating large-scale pretrained models through this lens reveals that they achieve strong generalization for certain concept types while still falling short of the ideal linear structure for others.

1 INTRODUCTION

Compositional understanding is the ability to combine simpler building blocks into novel, complex representations. It is widely regarded as a cornerstone of human intelligence (Dehaene et al., 2022). The Language of Thought hypothesis suggests that cognition arises from fundamental components and structured recombination rules (Fodor & Fodor, 1975). A growing body of work suggests that neural network representations often exhibit linear structure, where *concepts*, such as attributes or object properties in images, are represented as directions in the feature space (Park et al., 2023; Trager et al., 2023), and allow for arithmetic manipulations of them (Ravfogel et al., 2024; Wang et al., 2024b). Often referred to as the *linear representation hypothesis*, this idea holds promise for *explaining* many recent successes observed in compositional generalization (Trager et al., 2023; Abbasi et al., 2024; Mayilvahanan et al., 2024) and informing how structure can be exploited to improve compositional generalization.

We argue that without structured representations in a model $f = f_{\text{task}} \circ f_{\text{repr}}$, that can be seen as a composition of a feature extractor and task-head, the model may struggle with concept combinations it has not encountered during training, whether in zero-shot inference or adaptation. This is particularly concerning for downstream models that build atop of frozen representations, as these models must interpret the structure of feature-extractor’s representations even for unseen data to perform well.

Recent works have shown that features themselves are often *decodable*, and are capable to address spurious correlations (Rosenfeld et al., 2022; Kirichenko et al., 2023; Uselis & Oh, 2024). In particular, when a linear model is trained on frozen feature representations

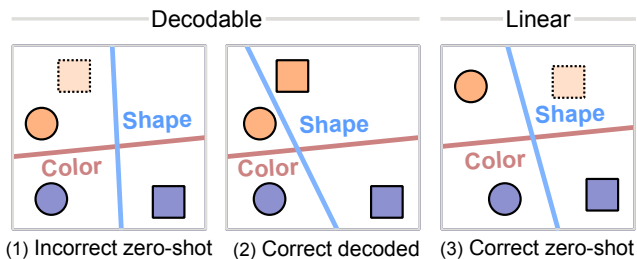


Figure 1: **Importance of linear feature structure for compositional generalization.** We illustrate a schematic for shape and color classification using linear models in a 2-dimensional feature space, comparing zero-shot and adapted cases with frozen feature extractor. (1) If the feature space lacks a **linear structure**, the model misclassifies the orange square in zero-shot inference. (2) Adaptation by adding orange square samples allows correct classification. (3) A **linearly structured** feature space enables correct zero-shot generalization without adaptation.

054 over a dataset with all possible concept combinations, the model often generalizes well within the
055 same data distribution. While this is promising, if we do not understand how the representation space
056 encodes concept combinations, ensuring generalization requires exposing the model to all possible
057 combinations, a task that quickly becomes infeasible. For example, as shown in Figure 1 (center),
058 adding datapoints that do not follow the structure the model expects can still enable correct classifica-
059 tion, indicating that adaptation can compensate for unstructured representations. However, collecting
060 such balanced datasets is often impractical, especially when the number of possible combinations is
061 large. If representations continue to exhibit a simple structure, such as *linearity*, even under unseen
062 concept combinations, generalization becomes possible without requiring exhaustive supervision, as
063 demonstrated in Figure 1 (right).

064 In this work, we investigate whether neural network feature representations can be decomposed as
065 the sum of independent, concept-specific vectors. Specifically, we show that this additive structure
066 naturally emerges when models are trained from scratch (Section 4), how it benefits compositional
067 generalization (Section 3.2), and that it is largely present in large-scale pre-trained models (Section
068 5).

070 2 RELATED WORK

071
072 Research on compositionality has taken several approaches, including both complexity-based and
073 structural perspectives (Elmoznino et al., 2025; Lepori et al., 2023). Neural networks often learn
074 to build complex representations by combining simple parts—a behavior sometimes attributed to
075 principles like Occam’s razor and the inherent simplicity observed in data (Ren & Sutherland,
076 2024; Geirhos et al., 2020; Valle-Pérez et al., 2018). Nonetheless, these models can sometimes rely
077 on misleading statistical patterns instead of capturing true compositional relationships (Pezeshki
078 et al., 2021; Scimeca et al., 2022), a problem that becomes especially apparent when certain valid
079 concept combinations are scarce in the training data (Sagawa et al., 2020). Other studies have
080 explored compositionality in generative models (Montero et al., 2022; 2020) or in settings with fixed
081 compositional datasets (Madan et al., 2021; Schott et al., 2022).

082 To promote better compositional generalization, recent methods have explored strategies like soft
083 prompting (Nayak et al., 2023), representation alignment (Wang et al., 2024a; Koishigarina et al.,
084 2025), and customized neural architectures (Zahran et al., 2024). There has also been notable progress
085 in object-centric approaches (Locatello et al., 2020; Wiedemer et al., 2023) and in developing metrics
086 for compositionality (Park et al., 2024b; Keysers et al., 2020), but mostly in language settings.

088 3 BENEFITS OF LINEARITY

090 3.1 STRUCTURE OF DATA AND MODELS

091 **Data structure.** At a high level, we study images that can be fully described by combinations of
092 discrete concepts (like color, shape, size). Each image maps to exactly one combination of concept
093 values, and each valid combination maps to exactly one image. We adapt the setup from (Trager et al.,
094 2023; Okawa et al., 2023; Park et al., 2024a) to study images that can be described by combinations
095 of concepts and their values.

096 **Definition 3.1** (Concepts and Concept Space). *A **concept space** $\mathcal{C} = \mathcal{C}_1 \times \dots \times \mathcal{C}_k$ is a Cartesian*
097 *product of k sets, where each set \mathcal{C}_i is called a **concept** and contains all possible values for concept*
098 *i . Each element $c_i \in \mathcal{C}_i$ is called a **concept value**, and each element $c \in \mathcal{C}$ represents a unique*
099 *combination of concept values (c_1, \dots, c_k) where $c_i \in \mathcal{C}_i$.*

100
101 For example, in the case of images, concepts could be the attributes of an image, such as color, shape,
102 and size, while concept values could be the specific color red, the shape triangle, and the size large.

103 We assume a mapping $c : \mathcal{X} \rightarrow \mathcal{C}$ that assigns to each image $\mathbf{x} \in \mathcal{X}$ its corresponding concept values
104 $c(\mathbf{x}) = (c_1, \dots, c_k) \in \mathcal{C}$. In other words, each image maps to exactly one combination of concept
105 values, and conversely, each valid combination of concept values maps to exactly one image. In this
106 work we assume all concepts are *discrete*, i.e. there is no inherent order between concept values of
107 any concept. In our experiments we will deviate from this assumption and work with concept spaces
where some concepts are ordinal, but we will treat them as discrete to keep this study simple.

Model structure. We consider feature extractors that map visual inputs to a representation space \mathbb{R}^d , which can be described as $f : \mathcal{X} \rightarrow \mathbb{R}^d$, where \mathcal{X} is the space of inputs and \mathbb{R}^d is the space of representations. This model can be extracted from most models trained under different settings: in supervised-learning models, this amounts to a linear layer on top of the features, in self-supervised models like DINO this corresponds to the encoder that compares augmented views of the same image, and in vision-language models like CLIP this represents the vision encoder that aligns visual features with text embeddings.

Representation structure. Feature spaces may exhibit structure in how they relate to the concept space \mathcal{C} . The feature extractor f maps images to representations that may encode concept information, and while the full complexity of this mapping can be difficult to analyze, we study whether the representations follow simple linear structure. In particular, we study how linearly concepts combine in the representation space - a property that is often assumed in concept learning and has emerging empirical support (Stein et al., 2024; Trager et al., 2023; Leemann et al., 2023). In particular, we study linear structure in the feature space, defined as follows:

Definition 3.2 (Linearly Factored Embeddings (Trager et al., 2023) and Concept Representations). *Given a concept space $\mathcal{C} = \mathcal{C}_1 \times \dots \times \mathcal{C}_k$, a collection of vectors $\{\mathbf{u}_c\}_{c \in \mathcal{C}}$ is linearly factored if there exist vectors $\mathbf{u}_{c_i} \in \mathbb{R}^d$ for all $c_i \in \mathcal{C}_i$ ($i = 1, \dots, k$), which we refer to as **concept representations**, such that for all $c = (c_1, \dots, c_k)$:*

$$\mathbf{u}_c = \mathbf{u}_{c_1} + \dots + \mathbf{u}_{c_k}. \quad (1)$$

Should such a linear mapping exist? Intuitively, we would want the representation to behave like a “switch” - having high similarity with vectors representing concepts that are present in the input, and low similarity with those that are not. This would allow detecting each concept independently of the others. As we show in Prop. A.1 in the Appendix, if representations satisfy an idealized version of this property, there must exist concept representations that combine linearly to form the full representation. While neural networks need not learn such representations, this provides one possible path to such linear structure emerging during training.

Task: Compositional generalization. We study how the structure of learned feature spaces enables compositional generalization in neural networks. Given a training dataset $\mathcal{D}_{\text{train}} \subset \mathcal{X} \times \mathcal{C}$ containing only a subset of possible concept combinations, we evaluate generalization to a test set $\mathcal{D}_{\text{test}} \subset \mathcal{X} \times \mathcal{C}$ containing novel combinations of familiar concepts. While images may contain many concepts, we focus on two target concepts whose ground truth labels we observe during training, *where each concept has at most n possible values*. Unlike standard i.i.d. generalization where train and test distributions match, here $\mathcal{D}_{\text{test}}$ contains systematically different pairings of these two n -valued concepts.

The key distinction from standard generalization is that test examples contain novel combinations of familiar concepts, rather than entirely new concepts. Success requires the model to learn representations that capture the compositional nature of the data rather than memorizing valid combinations.

3.2 BENEFITS OF LINEAR REPRESENTATIONS FOR COMPOSITIONAL GENERALIZATION

Recovering concept representations. Assuming that a representations from f are linearly factored, we can recover the individual concept vectors \mathbf{u}_i by observing just two combinations per concept value. Additionally, we can construct optimal classifiers of any concept value. Besides the linear factorization in the representations, the only condition we require is that the concept representations are not linearly-dependent. We summarize this result below.

Proposition 3.3 (Minimal Compositional Learning). *Let $f : \mathcal{X} \rightarrow \mathbb{R}^d$ be a feature extractor with linearly factored concept embeddings over \mathcal{C} . Let $U = \{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ and $V = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ be the concept vectors for the first and second concepts respectively, where $\text{span}(U + V)$ has dimension $2n - 1$. Suppose we only observe joint representations sharing concepts $c_i, c_j \in \{1, \dots, n\}$. Then $m = 2$ combinations per concept value suffice to learn a linear classifier that perfectly generalizes to all $(n - m) \cdot n$ unseen combinations.*

This theorem shows that with linear factorization, observing just two combinations per concept value allows recognizing all possible combinations. Specifically, we can generalize from $O(n)$ training combinations to all $O(n^2)$ possible combinations by decomposing and recombining concept vectors.

While this demonstrates the power of linear structure, it does not explain why and if networks would learn such representations.

3.3 EXPERIMENTAL SETUP

Training and testing sets $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$. In real-world scenarios, not all possible combinations of visual concept values occur with equal frequency: some combinations may be rare or entirely absent from the observed data. For any dataset, we focus on pairs of key concepts (e.g., color and shape). Our framework characterizes the number of possible training combinations through a parameter m which dictates the number of combinations each concept value appears with. For each concept value $i \in \{1, \dots, n\}$, we select m combinations with values from the other concept to form our training set. Within each valid training combination (each “cell” in our concept grid), we sample n_{cell} examples uniformly from all possible variations of the remaining unlabeled concepts $\mathcal{C}_{\text{vary}}$ (like position, orientation, background, etc.). This uniform sampling across $|\mathcal{C}_{\text{vary}}|$ possible variations ensures balanced representation of each concept combination across different visual contexts. We elaborate in the Appendix. We illustrate the training and test sets conceptually for $m = 3$ and $m = 2$ combinations per concept value in Figure 2.

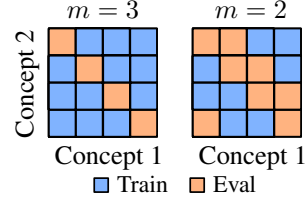


Figure 2: Training combinations for $n = 4$ concepts with $m = 3$ and $m = 2$ combinations per concept value. Blue cells indicate training combinations, while orange cells represent unseen test combinations. Each concept value appears in exactly m training combinations.

Throughout our experiments, we fix the training dataset size to be 40,000 samples, regardless of m , i.e. the diversity of concept combinations. In from-scratch case we perform oracle model selection (Gulrajani & Lopez-Paz, 2020) by picking the epoch that maximizes the sum of individual concept accuracies.

General evaluation approach. In this work, we evaluate whether linear structure emerges naturally. To do so, we train models from scratch (Section 4) and examine whether such structure is present in pre-trained models (Section 5).

In the from-scratch setup, we analyze how compositional generalization and the linearity of structure depend on data diversity, quantified by the factor m in Equation (2). In the pre-trained model setup, we instead assess accuracy under the assumption of linear representations. To achieve this, we construct optimal classifiers for each concept pair following Proposition 3.3. In this case, we always set $m = 2$, meaning only two concept combinations are observed for each concept value. Further details are provided in the respective sections.

Datasets. We use a set of five datasets for performing analysis in this work. We chose these datasets since they have associated concepts and their values associated with each sample. We elaborate in Appendix A.3.

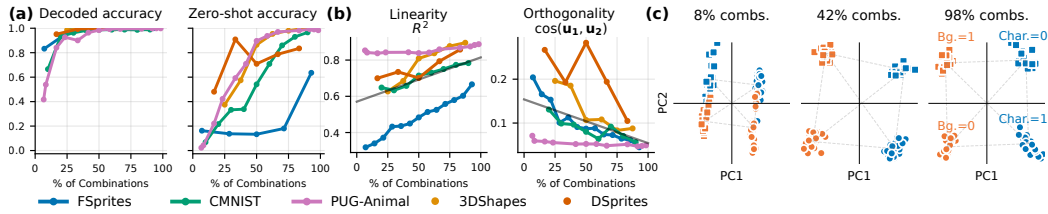
4 EMERGENT LINEAR STRUCTURE: THREE PHASES OF FEATURE LEARNING

In this section, we examine the zero-shot compositional generalization of models trained from scratch and its relation to decodability and linearity;.

Setup. We use a randomly-initialized RESNET-50 (He et al., 2015) with linear classification heads. The model outputs two predictions $f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))$ where $f_j : \mathcal{X} \rightarrow \mathcal{C}_j$ predicts the value of concept j using a shared backbone followed by separate linear heads. We learn fixed classification heads directly from visual data to provide an optimistic setting for compositional generalization. ViTs were considered but we found them underperform when trained from scratch compared to RESNET-50.

Metrics. We evaluate models on both compositional generalization and representation structure. For generalization, we measure zero-shot accuracy on unseen concept combinations in $\mathcal{C}_{\text{test}}$ (details in Appendix A.3). For representation structure, we analyze: (i) *Decodability* Kirichenko et al. (2023) - accuracy of linear probes trained on balanced data to assess concept discriminability, (ii) *Linearity* - R^2 score between representations and their linear reconstruction from concept components, and (iii)

216 *Orthogonality* - mean cosine similarity between concept representations. Detailed metric definitions
 217 and implementation are provided in Appendix A.2.
 218



219
220
221
222
223
224
225
226 **Figure 3: Linearity emerges with data diversity, while feature discriminability alone does not imply**
 227 **linear structure.** (a) Feature discriminability emerges early but does not imply compositional structure, (b)
 228 Linear concept representations only emerge with increased training diversity, as shown through R^2 scores and
 229 orthogonality measures. (c) PCA visualizations confirm evolution from entangled to linear feature organization
 230 as training diversity increases. X-axis represents percentage of training combinations m/n , with n being the
 231 maximum number of concept values.

232 **Results.** Our analysis reveals two key findings about how neural networks learn to represent concepts.
 233 First, we find that *linearity in representations* emerges naturally as models are exposed to more
 234 diverse training combinations. As shown in Figure 3(b), both the linear separability (R^2 scores) and
 235 orthogonality (cosine similarity) of concept dimensions improve with increased training diversity.
 236 This emergence of linear structure is accompanied by improved zero-shot generalization - Figure 3(a)
 237 shows that zero-shot accuracy on unseen combinations steadily increases as training diversity grows.

238 Second, we observe that this progression occurs in three distinct phases: (1) With limited concept
 239 combinations (0-10%), models learn spurious features with poor discrimination (decoded accuracy
 240 $< 80\%$) and random-level zero-shot performance, as shown by entangled representations in Figure 3(c)
 241 at 8%. (2) At moderate diversity (25-75%), linearity and orthogonality begin emerging (Figure 3(b)),
 242 with features becoming decodable (100% accuracy) and zero-shot performance reaching 60-80%. (3)
 243 At high diversity (75-100%), while discriminability plateaus, representations become strongly linear
 244 ($R^2 > 0.8$) and orthogonal (cosine similarity < 0.1), enabling zero-shot accuracy above 90% on the
 245 majority of the datasets. The PCA visualizations in Figure 3(c) qualitatively confirm this progression
 246 from entangled to linear organization.

247 These results indicates a link between training diversity and representation structure in NNs. While
 248 models can learn to discriminate individual concepts with limited data (at around 25%), linearity in
 249 representations emerges only with extensive concept diversity. Empirically, linearity and zero-shot
 250 accuracy appear to be directly related.

251
252 **5 DO LARGE PRE-TRAINED VISION MODELS EXHIBIT LINEAR**
 253 **REPRESENTATIONS?**
 254

255 Having established that linearity in representations emerges naturally when models are trained with
 256 sufficient concept diversity (Section 4), we now investigate whether modern pre-trained vision models
 257 exhibit similar properties. This question is particularly relevant given that these models are trained on
 258 massive, diverse datasets that should, in principle, expose them to many concept combinations. We
 259 evaluate several performant vision models to assess if their representations exhibit the linear structure
 260 needed for compositional generalization to novel concept combinations.

261 **Models.** We evaluate RESNET50-IMAGENET1K (He et al., 2015) for direct comparison purposes
 262 with from-scratch models, RESNET50-DINOv1 (Caron et al., 2021), for comparing pre-training
 263 data and training strategy impact, DINOv2 DINO-ViT-L/14 (Oquab et al., 2024) due to its strong
 264 performance in downstream tasks (Mamaghan et al., 2024), and CLIP-ViT-L/14 (Radford et al.,
 265 2021) which has demonstrated strong compositional capabilities across multiple studies (Abbasi
 266 et al., 2024; Stein et al., 2024; Oikarinen & Nguyen, 2023; Esfandiarpour et al., 2024).

267 **Metrics.** Following Proposition 3.3, we measure test accuracy using optimal classifiers constructed
 268 under the assumption of linearity of representations. These classifiers are derived from factored
 269 representations learned using only $m = 2$ combinations per concept value (details in Appendix. With
 them, we measure both the training and testing accuracies, as explained in Section 3.3.

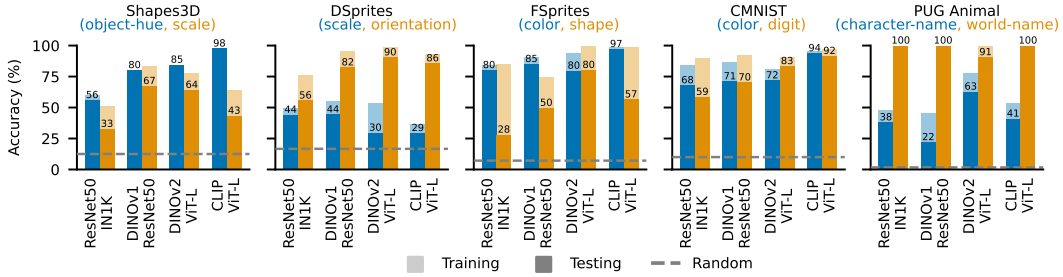


Figure 4: **Compositional generalization capabilities of pre-trained models when assuming the representations are compositional.** Bar plots show both training (transparent) and testing (solid) accuracy across different datasets (dSprites, Shapes3D, CMNIST, PUG-Animal) when using minimal training data ($k = 2$ combinations per concept) to learn linear concept representations for each concept. Dashed lines indicate random baseline performance. Following Proposition 3.3, we identified the factored representations \mathbf{u}_{c_1} and \mathbf{u}_{c_2} for each concept value using $m = 2$ combinations per concept value. While perfect generalization predicted by the proposition would require ideal linear compositionality, our empirical results show strong performance on certain concepts (e.g., $> 90\%$ accuracy on color, orientation, digit, and background concepts for either CLIP or DINOv2 models), with varying effectiveness across different concept types and models, suggesting that pre-trained representations exhibit partial linearity in their representations.

Results. The results are presented in Figure 4. Models achieve varying levels of accuracy across different concepts, consistently exceeding random chance (dashed lines) but never reaching perfect accuracy on all concept types. Some concept relationships appear inherently more complex to represent linearly - for instance, on DSprites, even the best model achieves only 50% training accuracy for scale classification.

Certain concept pairs show strong amenability to linear representation across all models. On PUG-Animal, all models achieve exceptionally high accuracy ($>90\%$) on background-character combinations, suggesting spatially separable concepts naturally induce more linear representations. The best model consistently exceeds 90% accuracy on *some* concept classification across all datasets. Additionally, models show clear specialization: CLIP excels at color-based tasks (highest accuracy on CMNIST color-digit and Shapes3D object-hue), while DINOv2 performs best on shape-based concepts (e.g. on scale, shape, orientation, and character).

While no model achieves the perfect generalization predicted by our theoretical analysis for ideally linear representations, these results demonstrate that pre-trained models exhibit linearity in their representations, varying in effectiveness across concept types. Strong performance on spatially separable concept pairs supports our hypothesis that linear representation organization facilitates compositional generalization.

6 CONCLUSION

In this work, we investigated how neural networks learn to represent and combine concepts by examining the relationship between feature representations and compositional generalization. We found that linearity in representations emerges naturally as training data diversity increases, but only after passing through distinct phases of spurious correlations and non-linear feature learning. As we argued and demonstrated, mere feature discriminability is insufficient for compositional generalization: models can learn to distinguish individual concepts when adapted to a full set of concept combinations, but fail to generalize to novel combinations zero-shot.

Building on these insights, we conducted an evaluation of whether large-scale pre-trained models like DINO and CLIP already exhibit compositional capabilities. By assuming linearity, we constructed optimal classifiers for each concept and evaluated their performance on unseen combinations. By analyzing their feature spaces through the lens of linearity, we found mixed results that varied across both model architectures and concept types. While DINO exhibited strong compositional capabilities for object-centric tasks and CLIP showed advantages for color-based generalization, neither achieved the perfect combinatorial generalization that our theoretical analysis shows is possible with ideally structured linear representations. These findings suggest that while current pre-training approaches can produce partially compositional features, there remains significant room for improvement in developing architectures and training objectives that more reliably develop linearity in representations.

324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377

REFERENCES

- Reza Abbasi, Mohammad Hossein Rohban, and Mahdieh Soleymani Baghshah. Deciphering the Role of Representation Disentanglement: Investigating Compositional Generalization in CLIP Models, July 2024. 1, 5
- Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers, May 2021. 5
- Stanislas Dehaene, Fosca Al Roumi, Yair Lakretz, Samuel Planton, and Mathias Sablé-Meyer. Symbols and mental programs: A hypothesis about human singularity. *Trends in Cognitive Sciences*, 26(9):751–766, September 2022. ISSN 13646613. doi: 10.1016/j.tics.2022.06.010. 1
- Eric Elmoznino, Thomas Jiralerspong, Yoshua Bengio, and Guillaume Lajoie. A Complexity-Based Theory of Compositionality, February 2025. 2
- Reza Esfandiarpour, Cristina Menghini, and Stephen H. Bach. If CLIP Could Talk: Understanding Vision-Language Model Representations Through Their Preferred Concept Descriptions, March 2024. 5
- Jerry A. Fodor and Jerry Alan Fodor. *The Language of Thought*. The Language and Thought Series. Crowell, New York, NY, 1975. ISBN 978-0-690-00802-9. 1
- Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut Learning in Deep Neural Networks. *Nature Machine Intelligence*, 2(11):665–673, November 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-00257-z. 2
- Ishaan Gulrajani and David Lopez-Paz. In Search of Lost Domain Generalization, July 2020. 4, 11
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition, December 2015. 4, 5, 11
- Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. Measuring Compositional Generalization: A Comprehensive Method on Realistic Data, June 2020. 2
- Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last Layer Re-Training is Sufficient for Robustness to Spurious Correlations, June 2023. 1, 4, 10
- Darina Koishigarina, Arnas Uselis, and Seong Joon Oh. CLIP Behaves like a Bag-of-Words Model Cross-modally but not Uni-modally, February 2025. 2
- Tobias Leemann, Michael Kirchhof, Yao Rong, Enkelejda Kasneci, and Gjergji Kasneci. When are Post-hoc Conceptual Explanations Identifiable?, June 2023. 3
- Michael A. Lepori, Thomas Serre, and Ellie Pavlick. Break It Down: Evidence for Structural Compositionality in Neural Networks, November 2023. 2
- Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-Centric Learning with Slot Attention, October 2020. 2
- Spandan Madan, Timothy Henry, Jamell Dozier, Helen Ho, Nishchal Bhandari, Tomotake Sasaki, Frédo Durand, Hanspeter Pfister, and Xavier Boix. When and how CNNs generalize to out-of-distribution category-viewpoint combinations, November 2021. 2
- Amir Mohammad Karimi Mamaghan, Samuele Papa, Karl Henrik Johansson, Stefan Bauer, and Andrea Dittadi. Exploring the Effectiveness of Object-Centric Representations in Visual Question Answering: Comparative Insights with Foundation Models, September 2024. 5
- Prasanna Mayilvahanan, Thaddäus Wiedemer, Evgenia Rusak, Matthias Bethge, and Wieland Brendel. Does CLIP’s Generalization Performance Mainly Stem from High Train-Test Similarity?, March 2024. 1

378 Milton L. Montero, Jeffrey S. Bowers, Rui Ponte Costa, Casimir J. H. Ludwig, and Gaurav Malhotra.
379 Lost in Latent Space: Disentangled Models and the Challenge of Combinatorial Generalisation,
380 April 2022. [2](#)

381 Milton Llera Montero, Casimir JH Ludwig, Rui Ponte Costa, Gaurav Malhotra, and Jeffrey Bowers.
382 The role of Disentanglement in Generalisation. In *International Conference on Learning*
383 *Representations*, October 2020. [2](#)

384 Nihal V. Nayak, Peilin Yu, and Stephen H. Bach. Learning to Compose Soft Prompts for Composi-
385 tional Zero-Shot Learning, April 2023. [2](#)

386 Tuomas Oikarinen and Lam M Nguyen. LABEL-FREE CONCEPT BOTTLENECK MODELS.
387 2023. [5](#)

388 Maya Okawa, Ekdeep Singh Lubana, Robert P. Dick, and Hidenori Tanaka. Compositional Abilities
389 Emerge Multiplicatively: Exploring Diffusion Models on a Synthetic Task, October 2023. [2](#)

390 Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre
391 Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Bal-
392 las, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat,
393 Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand
394 Joulain, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision,
395 February 2024. [5](#)

396 Core Francisco Park, Maya Okawa, Andrew Lee, Hidenori Tanaka, and Ekdeep Singh Lubana.
397 Emergence of Hidden Capabilities: Exploring Learning Dynamics in Concept Space, December
398 2024a. [2](#)

399 Kiho Park, Yo Joong Choe, and Victor Veitch. The Linear Representation Hypothesis and the
400 Geometry of Large Language Models, November 2023. [1](#)

401 Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The Geometry of Categorical and
402 Hierarchical Concepts in Large Language Models, June 2024b. [2](#)

403 Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and
404 Guillaume Lajoie. Gradient Starvation: A Learning Proclivity in Neural Networks, November
405 2021. [2](#)

406 Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal,
407 Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever.
408 Learning Transferable Visual Models From Natural Language Supervision, February 2021. [5](#)

409 Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan Cotterell. Linear Adversarial Concept
410 Erasure, December 2024. [1](#)

411 Yi Ren and Danica J. Sutherland. Understanding Simplicity Bias towards Compositional Mappings
412 via Learning Dynamics, September 2024. [2](#)

413 Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. Domain-Adjusted Regression or: ERM
414 May Already Learn Features Sufficient for Out-of-Distribution Generalization, October 2022. [1](#)

415 Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally Ro-
416 bust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case
417 Generalization, April 2020. [2](#)

418 Lukas Schott, Julius von Kügelgen, Frederik Träuble, Peter Gehler, Chris Russell, Matthias Bethge,
419 Bernhard Schölkopf, Francesco Locatello, and Wieland Brendel. Visual Representation Learning
420 Does Not Generalize Strongly Within the Same Domain, February 2022. [2](#)

421 Luca Scimeca, Seong Joon Oh, Sanghyuk Chun, Michael Poli, and Sangdoon Yun. Which Shortcut
422 Cues Will DNNs Choose? A Study from the Parameter-Space Perspective, February 2022. [2](#)

423 Adam Stein, Aaditya Naik, Yinjun Wu, Mayur Naik, and Eric Wong. Towards Compositionality in
424 Concept Learning, June 2024. [3](#), [5](#), [10](#)

432 Matthew Trager, Pramuditha Perera, Luca Zancato, Alessandro Achille, Parminder Bhatia, and
433 Stefano Soatto. Linear Spaces of Meanings: Compositional Structures in Vision-Language Models,
434 March 2023. 1, 2, 3, 13

435
436 Arnas Uselis and Seong Joon Oh. Intermediate Layer Classifiers for OOD generalization. In *The*
437 *Thirteenth International Conference on Learning Representations*, October 2024. 1

438 Guillermo Valle-Pérez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the
439 parameter-function map is biased towards simple functions. <https://arxiv.org/abs/1805.08522v5>,
440 May 2018. 2

441
442 Haoxiang Wang, Haozhe Si, Huajie Shao, and Han Zhao. Enhancing Compositional Generalization
443 via Compositional Feature Alignment, May 2024a. 2

444 Zihao Wang, Lin Gui, Jeffrey Negrea, and Victor Veitch. Concept Algebra for (Score-Based)
445 Text-Controlled Generative Models, February 2024b. 1, 10

446
447 Thaddäus Wiedemer, Jack Brady, Alexander Panfilov, Attila Juhos, Matthias Bethge, and Wieland
448 Brendel. Provable Compositional Generalization for Object-Centric Learning, October 2023. 2

449 Youssef Zahran, Gertjan Burghouts, and Yke Bauke Eisma. Anticipating Future Object Compositions
450 without Forgetting, July 2024. 2

451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485

486 A APPENDIX

487 A.1 TRAINING AND TESTING SETS CONSTRUCTION

488 Concretely, for each concept value $i \in \{0, \dots, n - 1\}$, we observe m combinations during training,
489 defining our training and test sets as:

$$490 \mathcal{C}_{\text{train}} := \bigcup_{i=1}^n \{(i, (i + j \bmod n)) : j \in \{0, \dots, m - 1\}\}, \mathcal{C}_{\text{test}} := (\mathcal{C}_1 \times \mathcal{C}_2) \setminus \mathcal{C}_{\text{train}}. \quad (2)$$

494 A.2 DETAILS ON METRICS IN FROM-SCRATCH MODELS

495 To quantify both compositional generalization capabilities and the underlying structure of learned
496 representations, we evaluate models using two complementary sets of metrics.

497 For generalization, we report zero-shot accuracy on $\mathcal{C}_{\text{test}}$, measuring the model’s ability to classify
498 unseen concept combinations. We report averaged accuracy for a concept pair considered (e.g., color
499 and shape, detailed in Appendix A.3).

500 For representation structure, we consider: (i) *Decodability*, following Kirichenko et al. (2023), we
501 train linear probes on balanced data and report average accuracy across concepts, indicating if features
502 capture concept information; that is, we merge the training and testing sets, and use a held-out dataset
503 covering all concept combinations for measuring decoded accuracy. (ii) *Linearity* - we compute
504 the coefficient of determination (R^2) between joint representations $\mathbf{f}(\mathbf{x})$ and their reconstruction
505 from individual concept representations $\sum_{i=1}^k \mathbf{u}_{c_i}$, where $R^2 = 1 - \frac{\sum_{\mathbf{x}} \|\mathbf{f}(\mathbf{x}) - \sum_{i=1}^k \mathbf{u}_{c_i}\|^2}{\sum_{\mathbf{x}} \|\mathbf{f}(\mathbf{x}) - \bar{\mathbf{f}}\|^2}$ with
506 $\bar{\mathbf{f}} = \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \mathbf{f}(\mathbf{x})$ measures how well representations follow linear structure. Here, \mathbf{f} represents the
507 mean representation across all samples. (iii) *Orthogonality* - we measure the mean cosine similarity
508 $\frac{1}{|\mathcal{C}_1||\mathcal{C}_2|} \sum_{i,j} \cos(\mathbf{u}_{c_i}, \mathbf{u}_{c_j})$ between concept representations to assess if concepts are encoded in
509 orthogonal subspaces, sometimes found in pretrained models (Stein et al., 2024; Wang et al., 2024b).

513 A.3 DATASETS

514 Table 1: Overview of datasets and their attributes. Numbers in parentheses indicate the cardinality $|\mathcal{C}_i|$ ($i = 1, 2$)
515 of possible values for each concept dimension.

516 Dataset	517 Attributes $\mathcal{C}_1, \mathcal{C}_2$ (Number of values per concept n)
518 PUG	519 Animal type (60), Background type (60)
520 Shapes3D	521 Scale (8), Object-hue (8)
522 DSprites	523 Scale (6), Orientation (6)
523 FSprites	524 Shape (14), Color (14)
524 Colored-MNIST	525 Digit (10), Color (10)

526 A.3.1 FUNNY SPRITES DATASET

527 We introduce the Funny Sprites dataset, an OOD dataset designed to test models’ ability to generalize
528 to previously unseen shape combinations. The dataset consists of sprites traced from 5-15 points
529 on a 128x128 pixel grid, creating a diverse set of abstract geometric shapes. This dataset serves as
530 an important test bed for evaluating compositional generalization, as it allows us to assess whether
531 models can recognize and combine novel shape features they have never encountered during training.
532

533 The sprites are generated by connecting traced points to form closed polygonal shapes, with variations
534 in:

- 535 • Shape (14 different base shapes)
- 536 • Scale (14 different sizes)
- 537 • Orientation (14 different angles)
- 538 • Position (14 x 14 grid positions)
- 539 • Color (14 distinct colors)

Each sprite can be dynamically recolored using a predefined palette of 14 colors, chosen to be visually distinct while maintaining good contrast. The dataset follows a similar structure to dSprites but introduces more complex geometric shapes to test generalization capabilities, ensuring that no previous model has seen such shapes. We illustrate the dataset with shape and orientation variations in Figure 5 in the case of $n = 14, m = 3$.

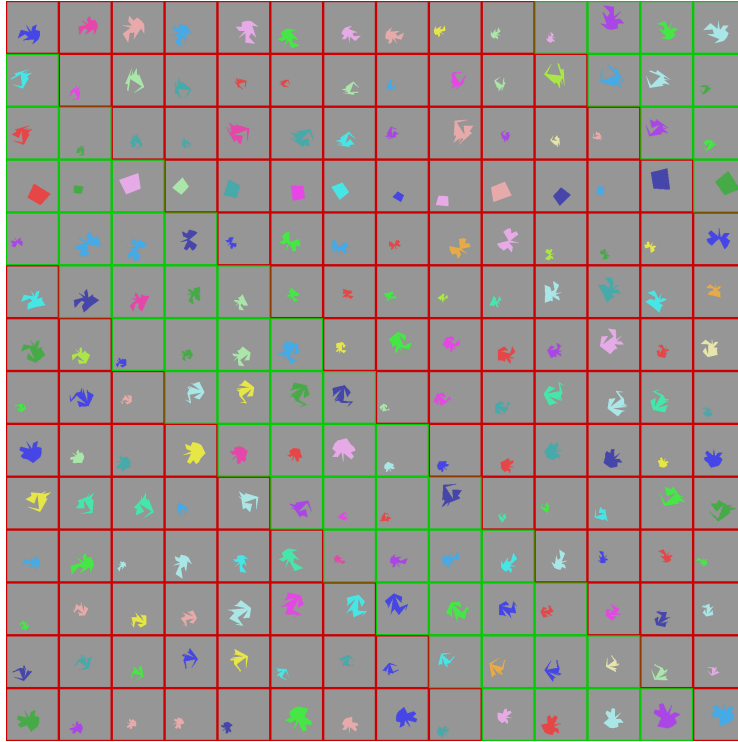


Figure 5: **Examples from the Funny Sprites dataset.** The figure shows different shape and orientation variations from our Funny Sprites dataset. Each sprite is generated by connecting 5-15 traced points to form unique geometric shapes. Here we show examples for $n = 14$ different values and $k = 2$ combinations of shape and orientation attributes.

A.4 DETAILS ON TRAINING AND EVALUATION

Model training. We use RESNET-50 (He et al., 2015) with linear classification heads. The model outputs two predictions $f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))$ where $f_j : \mathcal{X} \rightarrow \mathcal{C}_j$ predicts the value of concept j using a shared backbone followed by separate linear heads. We learn fixed classification heads directly from visual data to provide an optimistic setting for compositional learning through feature reuse. We found other baselines performing similarly to RESNET-50 but they were often slower. ViTs were considered but we found them underperform when trained from scratch compared to RESNET-50.

Model selection and metrics. For model selection, we use the average accuracy across all concepts at each epoch. We perform *oracle* model selection by directly evaluating models on the test set to select the best performing checkpoint (Gulrajani & Lopez-Paz, 2020). This allows us to focus on the fundamental capabilities of models rather than validation strategies.

A.5 IDEALIZED CLIP SETTING: TRAINING WITH CONCEPT SUPERVISION

Consider how CLIP’s training data naturally aligns with our concept structure framework. CLIP learns from a dataset $\mathcal{D}_{\text{data}} := \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ of image-text pairs, where each text description \mathbf{y}_i describes some subset of concepts present in image \mathbf{x}_i . Mathematically, we can express this as a text description function $d_y : 2^{\mathcal{C}} \rightarrow \mathcal{Y}$ that maps subsets of concept values to natural language

descriptions¹. For instance, given concept values $(c_1, c_2, c_3) = (\text{red}, \text{circle}, \text{large})$, d_y might output "a large red circle", "a red circle", or "a large circle" depending on which concepts are mentioned. During training, CLIP learns to align these partial concept descriptions with their corresponding images by maximizing the cosine similarity $\cos(f(\mathbf{x}), g(\mathbf{y})) = \frac{f(\mathbf{x})^\top g(\mathbf{y})}{\|f(\mathbf{x})\| \|g(\mathbf{y})\|}$ between matched image-text pairs while minimizing it for unmatched pairs. This training objective encourages the model to learn representations where any valid subset of concepts from an image has high alignment with the corresponding text embedding.

This training objective has important implications for how CLIP learns to structure its representation space. For any subset of concepts $S \subseteq c(\mathbf{x})$ present in an image \mathbf{x} , the cosine similarity between $f(\mathbf{x})$ and $g(d_y(S))$ should be higher than for any subset S' where some concepts from S are not matching the image. For example, given an image of a large red circle, the representation alignment with the text "a large red circle" should be higher than with "a small red circle" (incorrect concept).

More formally, for an image \mathbf{x} with concepts $c(\mathbf{x}) = (c_1, \dots, c_k)$ and any subset $S \subseteq \{c_1, \dots, c_k\}$:

$$\cos(f(\mathbf{x}), g(d_y(S))) > \cos(f(\mathbf{x}), g(d_y(S'))) \quad (3)$$

where S' is any subset that includes either (1) incorrect concept values for concepts in S , or (2) fewer concepts than S . This property encourages CLIP to learn representations where concept combinations are structured such that more complete and accurate descriptions have stronger alignment with the image representations that contain those concepts.

Assuming that CLIP produces constant cosine similarity values for any single concept value c_i , i.e., $\cos(f(\mathbf{x}), g(d_y(\{c_i\}))) = \alpha_{c_i}$ for some constant α_{c_i} that depends only on the concept value. Under this assumption, we can show that the representation space exhibits linear factorization.

Proposition A.1 (Fixed Cosine Similarities Admit Linear Factorization). *Assume that for every $x \in \mathcal{X}$ with $c(x) = (c_1, \dots, c_k)$ the following hold:*

(*) For each i ,

$$\cos(f(x), \mathbf{u}_{i,c_i}) = \frac{\langle f(x), \mathbf{u}_{i,c_i} \rangle}{\|f(x)\| \|\mathbf{u}_{i,c_i}\|} = \alpha_{i,c_i},$$

where α_{i,c_i} is a fixed constant (depending only on i and the active candidate c_i); for any other candidate $c \neq c_i$ in \mathcal{C}_i the cosine similarity takes some other fixed value.

(**) For every x ,

$$f(x) \in \text{span}\left(\bigcup_{i=1}^k \{\mathbf{u}_{i,1}, \dots, \mathbf{u}_{i,|\mathcal{C}_i|}\}\right) = \mathbb{R}^r, \quad \text{with } r := 1 + \sum_{i=1}^k (|\mathcal{C}_i| - 1).$$

Then, for every $x \in \mathcal{X}$ with configuration $c(x) = (c_1, \dots, c_k)$ there exist unique scalars $\beta_{1,c_1}, \dots, \beta_{k,c_k}$ such that

$$f(x) = \sum_{i=1}^k \beta_{i,c_i} \mathbf{u}_{i,c_i}, \quad \text{with } \beta_{i,c_i} = \frac{\langle f(x), \mathbf{u}_{i,c_i} \rangle}{\|\mathbf{u}_{i,c_i}\|^2}.$$

Moreover, if we further assume that $\|f(x)\| = C$ is constant for all x with the same configuration, then by (*)

$$\langle f(x), \mathbf{u}_{i,c_i} \rangle = \alpha_{i,c_i} C \|\mathbf{u}_{i,c_i}\|$$

so that

$$\beta_{i,c_i} = \alpha_{i,c_i} \frac{C}{\|\mathbf{u}_{i,c_i}\|}.$$

Proof. Since $f(x) \in \text{span}\{\mathbf{u}_{1,c_1}, \dots, \mathbf{u}_{k,c_k}\}$, there exist unique scalars β_{i,c_i} such that

$$f(x) = \sum_{i=1}^k \beta_{i,c_i} \mathbf{u}_{i,c_i}.$$

Taking the inner product with \mathbf{u}_{j,c_j} for any j gives

$$\langle f(x), \mathbf{u}_{j,c_j} \rangle = \beta_{j,c_j} \|\mathbf{u}_{j,c_j}\|^2,$$

¹When multiple instances of the same concept type appear (e.g., "three red circles"), the text description function can be extended to operate on multisets of concept values rather than just individual values. This preserves the mathematical framework while accommodating multiplicity.

648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701

so that

$$\beta_{j,c_j} = \frac{\langle f(x), \mathbf{u}_{j,c_j} \rangle}{\|\mathbf{u}_{j,c_j}\|^2}.$$

Then by (*) we have

$$\langle f(x), \mathbf{u}_{j,c_j} \rangle = \alpha_{j,c_j} \|f(x)\| \|\mathbf{u}_{j,c_j}\|.$$

If $\|f(x)\| = C$, it follows that

$$\beta_{j,c_j} = \alpha_{j,c_j} \frac{C}{\|\mathbf{u}_{j,c_j}\|}.$$

□

We can absorb the coefficients into the concept representations by defining

$$\tilde{\mathbf{u}}_{i,c_i} := \beta_{i,c_i} \mathbf{u}_{i,c_i} = \alpha_{i,c_i} \frac{C}{\|\mathbf{u}_{i,c_i}\|} \mathbf{u}_{i,c_i}.$$

This allows us to express $f(x)$ as a simple sum of these modified concept vectors for every x with configuration $c(x) = (c_1, \dots, c_k)$, we have

$$f(x) = \sum_{i=1}^k \tilde{\mathbf{u}}_{i,c_i}.$$

A.6 DETAILS ON THE SUPERVISED TRAINING PROCEDURE

Following (Trager et al., 2023), we list a property of linearly factored embedding:

Proposition A.2 (Unique Mean-Centered Decomposition). *For any linearly factored embeddings $\{\mathbf{f}_c\}_{c \in \mathcal{C}}$, there exist unique concept value embeddings $\{\mathbf{u}_{c_i}\}_{c_i \in \mathcal{C}_i}$ for each concept i with zero mean ($\sum_{c_i \in \mathcal{C}_i} \mathbf{u}_{c_i} = \mathbf{0}$), such that:*

$$\mathbf{f}_c = \mathbf{u}_0 + \mathbf{u}_{c_1} + \dots + \mathbf{u}_{c_k} \quad \text{where } \mathbf{u}_{c_i} = \frac{1}{|\mathcal{C}_i|} \sum_{c \in \mathcal{C}_i} \mathbf{f}_c \quad (4)$$

where \mathbf{u}_0 is the sum of the means of the representations: $\mathbf{u}_0 = \sum_{i=1}^k \frac{1}{|\mathcal{C}_i|} \sum_{c_i \in \mathcal{C}_i} \mathbf{f}_{c_i}$. Additionally, each \mathbf{u}_{c_i} can be recovered by taking the mean over centered representations \mathbf{f}_c that contain concept value c_i :

$$\mathbf{u}_{c_i} = \frac{1}{|\{c \in \mathcal{C} : c_i \in c\}|} \sum_{c \in \mathcal{C} : c_i \in c} (\mathbf{f}_c - \frac{1}{|\mathcal{C}|} \sum_{c' \in \mathcal{C}} \mathbf{f}_{c'}) \quad (5)$$

This proposition essentially tells us that if we recover any decomposition of linearly factored embeddings $\{\mathbf{f}_c\}_{c \in \mathcal{C}}$, the centered components of the decomposition are unique and match those of the factored embeddings.

In practise, we often do not have access to the observations of the concept values c_i for each $\mathbf{x} \in \mathcal{X}$. A more realistic assumption is that we have access to only a subset of concepts $\mathcal{S} \subset \mathcal{C}$. We can thus define the pairwise joint embedding between concept values in \mathcal{S} as follows.

Definition A.3 (Pairwise Joint Embedding). *Given a concept space $\mathcal{C} = \mathcal{C}_1 \times \dots \times \mathcal{C}_k$, the pairwise joint embedding between concepts $i, j \in \{1, \dots, k\}$ and their values $c_i, c_j \in \mathcal{C}_i, \mathcal{C}_j$ is defined as:*

$$\mathbf{u}_{c_i,c_j} = \frac{1}{|\{\mathbf{x} \in \mathcal{D} : c(\mathbf{x})_i = c_i, c(\mathbf{x})_j = c_j\}|} \sum_{\mathbf{x} \in \mathcal{D} : c(\mathbf{x})_i = c_i, c(\mathbf{x})_j = c_j} f(\mathbf{x}). \quad (6)$$

It then immediately follows that the pairwise joint embedding between concepts $i, j \in \{1, \dots, k\}$ is equal to the sum of individual concept embeddings:

Lemma A.4. *Under linearly factored embeddings, the pairwise joint embedding \mathbf{u}_{c_i,c_j} is equal to the sum of individual concept embeddings:*

$$\mathbf{u}_{c_i,c_j} = \mathbf{u}_{c_i} + \mathbf{u}_{c_j} \quad (7)$$

where \mathbf{u}_{c_i} and \mathbf{u}_{c_j} are the factored representations of concepts i and j respectively.

702 *Proof.* Note that

$$703 \mathbf{u}_{c_i, c_j} = \frac{1}{|\{\mathbf{x} \in \mathcal{D} : c(\mathbf{x})_i = c_i, c(\mathbf{x})_j = c_j\}|} \sum_{\mathbf{x} \in \mathcal{D} : c(\mathbf{x})_i = c_i, c(\mathbf{x})_j = c_j} f(\mathbf{x}) \quad (8)$$

$$704 = \frac{1}{|\{\mathbf{x} \in \mathcal{D} : c(\mathbf{x})_i = c_i, c(\mathbf{x})_j = c_j\}|} \sum_{\mathbf{x} \in \mathcal{D} : c(\mathbf{x})_i = c_i, c(\mathbf{x})_j = c_j} \sum_{l=1}^k \mathbf{u}_{c_l} \quad (9)$$

$$705 = \mathbf{u}_{c_i} + \mathbf{u}_{c_j} + \sum_{l \notin \{i, j\}} \frac{1}{|\mathcal{C}_l|} \sum_{c_l \in \mathcal{C}_l} \mathbf{u}_{c_l} \quad (10)$$

$$706 = \mathbf{u}_{c_i} + \mathbf{u}_{c_j} \quad (11)$$

707 where the second line expresses $f(\mathbf{x})$ as the sum of all concept embeddings from the linear factorization, the third line separates out the fixed concepts c_i and c_j and averages over all other concepts, and the final equality follows since $\sum_{c_l \in \mathcal{C}_l} \mathbf{u}_{c_l} = \mathbf{0}$ for any concept dimension l by Proposition A.2. \square

708 **Theorem A.5** (Minimal Compositional Learning). *Let $f : \mathcal{X} \rightarrow \mathbb{R}^d$ be a feature extractor with linearly factored concept embeddings over \mathcal{C} . Let $U = \{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ and $V = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ be the concept vectors for the first and second concepts respectively, where $\text{span}(U + V)$ has dimension $2n - 1$. Suppose we only observe joint representations sharing concepts $c_i, c_j \in \{1, \dots, n\}$. Then $m = 2$ combinations per concept value suffice to learn a linear classifier that perfectly generalizes to all $(n - m) \cdot n$ unseen combinations.*

709 *Proof.* We show this in two steps. The proof relies on establish that the joint factored embeddings under the training data are identifiable. Then, we show that the number of independent equations is equal to the number of unknowns, and that every equation provides independent information about the factored representations. Additionally, we assume that the restrictions are placed only on the concepts on which the target depends; we assume that all other concept combinations are present in the training data.

710 **Part 1: Identifying joint factored embeddings $\mathbf{u}_{c_1^i, c_2^j}$.**

711 We assume $k = 2$ for simplicity, but the same applies for higher k . First, note that we observe the following combinations:

$$712 \mathcal{C}_{\text{train}} = \{(i, i) : i \in [n]\} \cup \{(i, i + 1) : i \in [n - 1]\} \cup \{(n, 1)\} \quad (12)$$

$$713 = \{(1, 1), (2, 2), \dots, (n, n)\} \cup \{(1, 2), (2, 3), \dots, (n - 1, n)\} \cup \{(n, 1)\} \quad (13)$$

714 with $|\mathcal{C}_{\text{train}}| = n + (n - 1) + 1 = 2n$ total combinations. This dataset is restricted to the combinations in $\mathcal{C}_{\text{train}}$, but varies in other concepts. We denote this dataset as $\mathcal{D}_{\text{train}} := \{(c_1, c_2, \mathbf{x}) : (c_1, c_2) \in$

$\mathcal{C}_{\text{train}}, \mathbf{x} \in \mathcal{X}$. The average representation over these training combinations is:

$$\bar{\mathbf{u}}_{\text{train}} = \frac{1}{2n} \left(\sum_{i=1}^n \mathbb{E}_{\mathbf{x}: i \in c_1(\mathbf{x}), i \in c_2(\mathbf{x})} [f(\mathbf{x})] + \sum_{i=1}^{n-1} \mathbb{E}_{\mathbf{x}: i \in c_1(\mathbf{x}), i+1 \in c_2(\mathbf{x})} [f(\mathbf{x})] + \mathbb{E}_{\mathbf{x}: n \in c_1(\mathbf{x}), 1 \in c_2(\mathbf{x})} [f(\mathbf{x})] \right) \quad (14)$$

$$= \frac{1}{2n} \left(\sum_{i=1}^n (\mathbf{u}_{c_1^i, c_2^i} + \mathbf{f}) + \sum_{i=1}^{n-1} (\mathbf{u}_{c_1^i, c_2^{i+1}} + \mathbf{f}) + (\mathbf{u}_{c_1^n, c_2^1} + \mathbf{f}) \right) \quad (\text{since } \mathbf{u}_{c_1, c_2} = f(\mathbf{x}) - \mathbf{f}) \quad (15)$$

$$= \frac{1}{2n} \left(\sum_{i=1}^n (\mathbf{u}_{c_1^i} + \mathbf{u}_{c_2^i} + \mathbf{f}) + \sum_{i=1}^{n-1} (\mathbf{u}_{c_1^i} + \mathbf{u}_{c_2^{i+1}} + \mathbf{f}) + (\mathbf{u}_{c_1^n} + \mathbf{u}_{c_2^1} + \mathbf{f}) \right) \quad (16)$$

$$= \frac{1}{2n} \left(\sum_{i=1}^n \mathbf{u}_{c_1^i} + \sum_{i=1}^{n-1} \mathbf{u}_{c_1^i} + \mathbf{u}_{c_1^n} + \sum_{i=1}^n \mathbf{u}_{c_2^i} + \sum_{i=1}^{n-1} \mathbf{u}_{c_2^{i+1}} + \mathbf{u}_{c_2^1} + 2n\mathbf{f} \right) \quad (17)$$

$$= \frac{1}{2n} \left(2 \sum_{i=1}^n \mathbf{u}_{c_1^i} + 2 \sum_{i=1}^n \mathbf{u}_{c_2^i} + 2n\mathbf{f} \right) \quad (18)$$

$$= \frac{1}{2n} (2 \cdot \mathbf{0} + 2 \cdot \mathbf{0} + 2n\mathbf{f}) \quad (\text{since } \sum_{i=1}^n \mathbf{u}_{c_1^i} = \sum_{i=1}^n \mathbf{u}_{c_2^i} = \mathbf{0}) \quad (19)$$

$$= \mathbf{f} \quad (20)$$

Thus, we can identify the factored representations $\mathbf{u}_{c_1^i}$ and $\mathbf{u}_{c_2^i}$ for each concept value $i \in [n]$ from the training data since the average representation over the training data under our training dataset is the global mean embedding \mathbf{f} . With this, we can compute $\mathbf{u}_{c_1^i, c_2^i}$ for $2n$ combinations.

Part 2: Identifying the individual factored representations $\mathbf{u}_{c_1^i}$ and $\mathbf{u}_{c_2^i}$ for each concept value $i \in [n]$.

Consider a training set with exactly two combinations per concept value. By the linear factorization property, for any combination (i, j) in our training set, we have: $\mathbf{u}_{c_1^i, c_2^j} = \mathbf{u}_{c_1^i} + \mathbf{u}_{c_2^j}$, where c_1^i denotes value i for the first concept and c_2^j denotes value j for the second concept.

Let $\mathbf{U}_1, \mathbf{U}_2 \in \mathbb{R}^{d \times n}$ be matrices whose columns are the unknown factored representations $\mathbf{u}_{c_1^i}$ and $\mathbf{u}_{c_2^i}$ respectively for $i \in [n]$. Let $\mathbf{V} \in \mathbb{R}^{d \times 2n}$ be the matrix of observed pairwise joint embeddings $\mathbf{u}_{c_1^i, c_2^j}$ for the $2n$ training combinations. The system of equations can be written as:

$$\underbrace{\begin{bmatrix} \mathbf{u}_{c_1^1, c_2^1} \\ \mathbf{u}_{c_1^1, c_2^2} \\ \vdots \\ \mathbf{u}_{c_1^n, c_2^n} \\ \mathbf{u}_{c_1^1, c_2^2} \\ \mathbf{u}_{c_1^2, c_2^3} \\ \vdots \\ \mathbf{u}_{c_1^{n-1}, c_2^n} \\ \mathbf{u}_{c_1^n, c_2^1} \end{bmatrix}}_{\mathbf{V}} = \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 & | & 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & | & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & | & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & | & 0 & 0 & \cdots & 1 \\ \hline 1 & 0 & \cdots & 0 & | & 0 & 1 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & | & 0 & 0 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & | & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & | & 0 & 0 & \cdots & 1 & 0 \\ 1 & 0 & \cdots & 0 & | & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}}_{\mathbf{U}_1} \underbrace{\begin{bmatrix} \mathbf{u}_{c_1^1} \\ \mathbf{u}_{c_1^2} \\ \vdots \\ \mathbf{u}_{c_1^n} \\ \mathbf{u}_{c_2^1} \\ \mathbf{u}_{c_2^2} \\ \vdots \\ \mathbf{u}_{c_2^n} \end{bmatrix}}_{\begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{bmatrix}} \quad (21)$$

We note that this system is full rank, as the design matrix has linearly independent rows. The first block of rows corresponds to the diagonal combinations (i, i) , while the second block corresponds to cyclic combinations $(i, i + 1)$ (with wraparound from n to 1). These form distinct patterns that ensure linear independence.

Given this full rank system with $2n$ equations and $2n$ unknowns (the factored representations $\mathbf{u}_{c_1^i}$ and $\mathbf{u}_{c_2^j}$ for each concept value), we can uniquely solve for the factored concept embeddings. For $k > 2$ combinations per concept value, we get more equations while maintaining the same number of unknowns, making the system overdetermined and the solution more robust.

Once we recover these factored representations, we can compute $\mathbf{u}_{c_1^i, c_2^j} = \mathbf{u}_{c_1^i} + \mathbf{u}_{c_2^j}$ for any combination (i, j) , including the $(n - 2)n$ unseen ones.

Part 3: Optimality of classifiers. To show that we can construct classifiers that provably generalize to novel combinations, we simply note that by assumption no concept representation is within the span of remaining representations. As such, given $U := \text{span}(\{\mathbf{u}_i\}_{i=1}^{|\mathcal{C}_1|})$, and $V := \text{span}(\{\mathbf{v}_i\}_{i=1}^{|\mathcal{C}_2|})$, such that $\dim(U) = |\mathcal{C}_1| - 1$ and $\dim(V) = |\mathcal{C}_2| - 1$ and $U \cap V = \{0\}$, any vector \mathbf{w} in their joint span can be uniquely decomposed as $\mathbf{w} = \mathbf{u} + \mathbf{v}$ where $\mathbf{u} \in U$, $\mathbf{v} \in V$ and $\mathbf{u} \perp \mathbf{v}$. This allows us to construct projection matrices P_U and P_V onto these orthogonal subspaces, which can then be used to build optimal classifiers by projecting input features onto the respective concept dimensions.

□

We note that the proof above is constructive, and can be used to recover the factored representations from the training data. We summarize the steps in the Algorithm 1. In practise, since we do not observe all possible combinations of unlabeled concepts, we use empirical approximations of the expectations.

Algorithm 1 Recovering Factored Concept Representations for $k = 2$ Concepts

Require: Training dataset $\mathcal{D}_{\text{train}}$ where each individual concept appears in at least 2 different combinations ($k \geq 2$)

Require: Feature extractor $f : \mathcal{X} \rightarrow \mathbb{R}^d$

Ensure: Factored concept representations $\{\mathbf{u}_{c_1^i}\}_{i=1}^n, \{\mathbf{u}_{c_2^j}\}_{i=1}^n$

- 1: Compute global mean embedding: $\mathbf{f}_d \leftarrow \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{\mathbf{x} \in \mathcal{D}_{\text{train}}} f(\mathbf{x})_d$ for each dimension d
 - 2: **for** $d = 1$ to d **do**
 - 3: Initialize design matrix $\mathbf{A} \in \mathbb{R}^{2n \times 2n}$ based on observed combinations
 - 4: Initialize $\mathbf{v} \in \mathbb{R}^{2n}$ to store joint embeddings for dimension d
 - 5: $row \leftarrow 1$
 - 6: **for** each combination (i, j) in training set **do**
 - 7: $\mathbf{u}_{c_1^i, c_2^j} \leftarrow \frac{1}{|\{\mathbf{x}: c(\mathbf{x})_1=i, c(\mathbf{x})_2=j\}|} \sum_{\mathbf{x}: c(\mathbf{x})_1=i, c(\mathbf{x})_2=j} f(\mathbf{x})_d - \mathbf{f}_d$
 - 8: Store $\mathbf{u}_{c_1^i, c_2^j}$ in position row of \mathbf{v}
 - 9: Update row row of \mathbf{A} with indicators for concepts i and j
 - 10: $row \leftarrow row + 1$
 - 11: **end for**
 - 12: Solve system $\mathbf{A} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} = \mathbf{v}$ for dimension d
 - 13: Store solutions in $\{\mathbf{u}_{c_1^i}\}_{i=1}^n, \{\mathbf{u}_{c_2^j}\}_{i=1}^n$ at dimension d
 - 14: **end for**
 - 15: **return** $\{\mathbf{u}_{c_1^i}\}_{i=1}^n, \{\mathbf{u}_{c_2^j}\}_{i=1}^n$
-