Understanding self-supervised learning using controlled datasets with known structure

Katherine L. Hermann[*] **Ting Chen, Mohammad Norouzi, Simon Kornblith**
Google Research, Brain Team

Abstract

Modern self-supervised learning techniques such as SimCLR can learn surprisingly good representations from unlabeled images. However, little is known regarding how the learned representations are organized. Here, we propose to investigate the representations learned by contrastive self-supervised models on synthetically generated datasets where all structure is known. We investigate how SimCLR represents data with discrete, hierarchically structured classes, showing that the distances between examples in the representation space reflect the hierarchy. We also probe the representation of continuous appearance parameters. We demonstrate that the representations learned in a simplified setting capture several of the properties of representations learned from ImageNet, and study systematically the interactions between augmentations and continuous features. Our experiments show that contrastive learning methods exhibit non-trivial behavior even on simple datasets where the origins of this behavior may be more tractably investigated.

1 Introduction

Contrastive methods for self-supervised learning [34, 22, 17, 35, 14, 1, 32, 21, 12, 7, 8] currently achieve state-of-the-art performance on several computer vision tasks. These methods use a contrastive loss to encourage representations of different views (augmentations) of the same image to be similar, and representations of different images to be dissimilar. Questions remain about why contrastive representation learning supports transfer to tasks like ImageNet classification. Given that two views of an image share many similarities, why should contrastive learning lead to representations of task-relevant latent structure?

Here, we examine how SimCLR [7] learns two kinds of structure: discrete class labels organized in a semantic hierarchy, and continuous appearance parameters such as position and color. In both cases, we use synthetic datasets where all structure is known by construction. These datasets support controlled experiments that are not possible with natural images. Our setup permits investigation of important practical components of SimCLR, such as the use of the input to an MLP projection head at the end of the network as the embedding for downstream tasks, and of temperature-scaled cross-entropy loss on L^2-normalized vectors.

In our study of discrete class structure, we find that SimCLR naturally learns to organize its inputs according to their hierarchical relationships: like supervised models [27, 16, 3], contrastive self-supervised models learn an embedding space in which items that differ in the top levels of the hierarchy are most distinctly separated, with separation decreasing further down. The temperature parameter of the normalized cross-entropy loss [31] influences the degree to which the model resolves items at the hierarchy's lower levels.

To study the representation of continuous appearance parameters, we construct a variant of the `dSprites` dataset [20] in which images consist of a 2-dimensional shape that varies in color, scale,

[*] Corresponding email: `hermannk@stanford.edu`. This work was done as part of a Google internship.

Self-Supervised Learning: Theory and Practice workshop, NeurIPS 2020

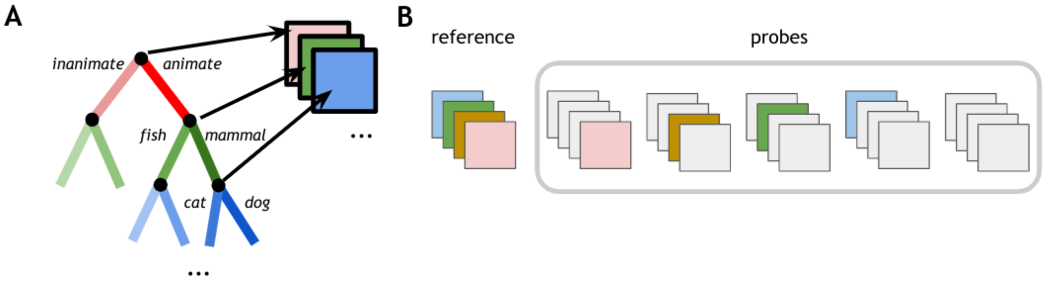

Figure 1: **(A)** In the hierarchical dataset, input channels corresponded to levels of a "semantic" hierarchy. **(B)** In the *channel preference analysis*, probes matched a reference item in exactly one channel, or in no channels (baseline). To determine the contrastive model's feature preferences, we measured the cosine similarity between a reference item and each probe.

and position. Despite its simplicity, we find that this dataset captures behaviors characteristic of contrastive learning on ImageNet. The simplified dataset allows us to isolate the effects of augmentations on particular features. We consider a *congruent* case in which, for an augmentation–feature pair, the augmentation changes the feature value (e.g. color distortion–color), and an *incongruent* one in which it does not (e.g. random crops–color) (see Figure B.4). Latent space visualization allows us to examine the evolution of feature representations through SimCLR's projection head layers.

2 How does a contrastive model learn class-relevant features?

Previous work [27, 28, 26, 16, 25, 3] has shown that supervised models progressively differentiate hierarchical structure in their input data over training. The task of a contrastive model like SimCLR is to compute the features that determine whether two views represent the same object. If the model uses class features, it can reuse them in support of judgments about many images (all images from a given class). So, class-level features are perhaps the analog of the most useful "rough-cut" distinctions in the supervised case, and we hypothesized that a contrastive would learn them first.

Dataset. In our hierarchical dataset, channels of an input correspond to different levels of a "semantic" hierarchy (Figure 1). The i^{th} channel of each input is one of $\prod_{j=0}^{i-1} b_j$ noise masks, where b_j is the branching factor of the j^{th} level of the hierarchy, and each noise mask is 38×38 px, with each pixel drawn from a uniform distribution. We present results for a dataset with a hierarchy of four levels, each with branching factor 10. Since all masks are drawn from the same distribution, none should be inherently easier to learn than any other; this ensures that any preference the model displays for one level over another is due to its position in the hierarchy rather than to its visual features.

Model. We trained a model (see Appendix A.1) to minimize normalized temperature-scaled cross-entropy loss [31, 7] for 100 epochs, using a batch size of 256 and an Adam optimizer with learning rate of 0.001. Augmentations consisted of 28×28 px random crops without resizing. We held the dataset and initialization fixed, but varied the temperature, across experiments.

Feature preference analyses. In a *channel preference analysis*, we constructed probe items that matched a reference item in exactly one channel (Figures 1 and B.1). We note that natural items could not have this property, since if two items belong to the same subclass (e.g. dog), they necessarily belong to the same superclass (animal). Over training, for each projection head layer, we measured the cosine similarity of layer activations in response to a reference item and each of its probes (4, one matching each channel). In one version of the analysis, reference items (200) were sampled from the training data and probe item channels were sampled from masks seen during training. In another version, reference items (200) and probe channels were populated with randomly drawn masks. See Appendix A.1 and Supplementary Figures B.2, and B.3 for a second feature preference analysis.

Results. We found that the contrastive model learned hierarchical structure from the data. At each projection head layer, it learned an embedding space such that a reference item was most similar to probes that matched its class at the top level of the hierarchy, and least similar to probes that matched its class at only the bottom level (Figures 2 and B.1). When the contrastive loss temperature was low (0.1), similarity dropped off in a graded way as a function of the hierarchy level shared with the reference item. A second analysis (Figures B.2 and B.3) recapitulated this general finding: when similarity was analyzed in a hierarchically constrained way (Appendix A.1), items that differed in top-level class were well separated, whereas items that differed only in lower-level classes exhibited

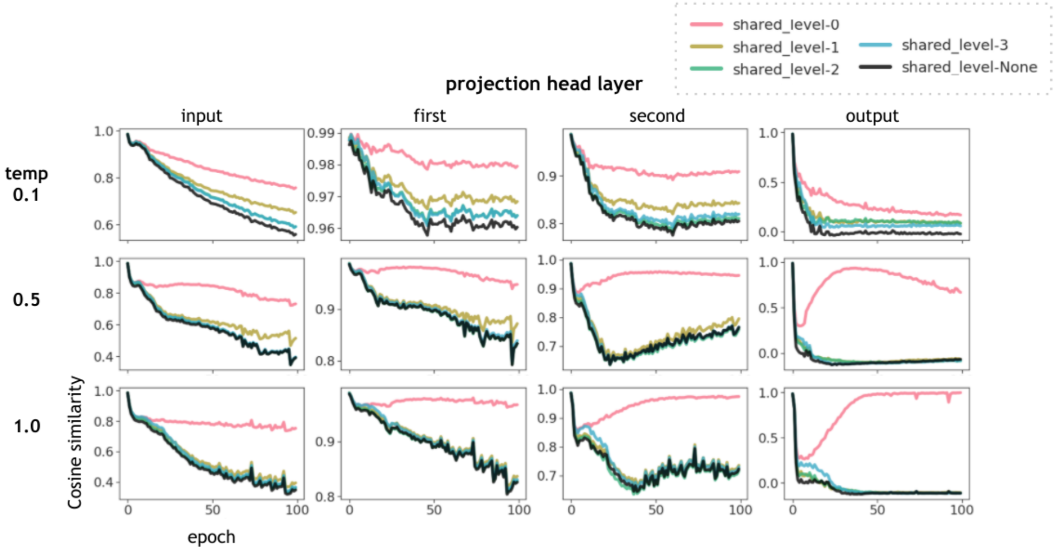

Figure 2: SimCLR models learn hierarchical structure. Cosine similarity of reference items to probes drawn from the train set in models trained with different contrastive loss temperatures (rows), through layers of the projection head (columns). Similarity is highest when the probe matches the reference item in terms of class at the top level of the hierarchy (pink). Black indicates baseline similarity for probes that did not match reference items on any channel. The contrastive loss temperature influences how subsequent hierarchy levels are privileged.

less separation. Again, the contrastive loss temperature controlled the resolution of intermediate-level structure. Although previous work has attempted to explain the success of contrastive learning through maximization of mutual information [22, 17, 23, 32], this framework cannot explain why SimCLR learns hierarchy in this setting: The top-level class is fully determined by the bottom-level class, and thus provides no additional information.

3 How do contrastive models represent continuous appearance parameters?

In addition to containing class structure, natural images contain continuously varying latent factors like color and object size. Extensive research has explored how unsupervised models like VAEs can discover disentangled representations of these factors [2, 15, 33, 18, 5, 10, 6, 9, 24, 19]. What does a contrastive model like SimCLR learn?

Dataset. We used a modified version of `dSprites` [20] consisting of 15,360 64×64 px RGB images. All images contained the same object at a fixed orientation against a black background, but the object varied across images in color (10 hues sampled uniformly from HUSL colorspace following [19]), scale (6 scales), and position (16 x-pos. × 16 y-pos., for 256 (x, y) positions); see Figure B.4. We created a full cross of the features.

Models. We trained four ResNet-18 architectures on the SimCLR objective [7] (see Appendix A.2), varying the type of augmentation applied during training: "no augmentation", "color jitter" (a combination of color jitter and color dropping, see Appendix A.2), "random crop" (random resized crops, see Appendix A.2), and "crop + jitter" (both color jitter and random crop). We considered the untrained model as a baseline.

Feature decodability experiments. To determine to what extent a model represents each continuous appearance parameter (color, scale, position), we trained and evaluated linear decoders to read out color, scale, or position from each layer of the projection head given images seen during training. A linear decoder consisted of a single linear layer trained to minimize softmax cross-entropy loss. Decoder inputs were layer activations (including the ReLU) from a frozen, trained model in response to an image, and decoder targets were class labels corresponding to a particular feature (color, scale, or position). When decoding color, we assessed decoder performance on a validation set containing held-out scales (2) and positions (10%). When decoding scale and position, we used a validation set containing held-out colors (2). See Appendix A.2 for additional details.

Results. To confirm the validity of studying contrastive models in our simplified setup, we compared feature decodability through the projection head with results reported for a model trained on ImageNet. We found that, as on ImageNet [7], appearance parameters were more decodable early in the projection

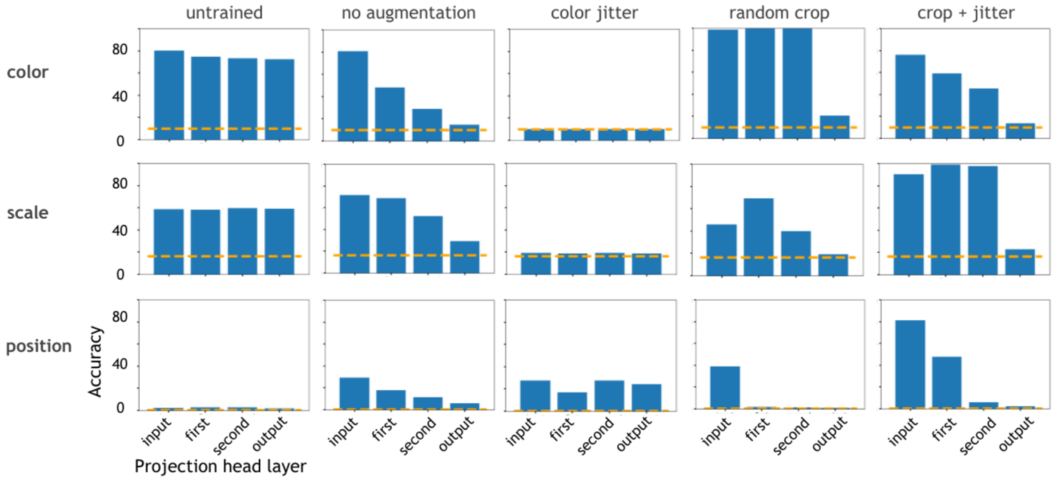

Figure 3: Decodability of object color (chance = 1/10), scale (1/6), and position (1/256) (rows) from SimCLR models (untrained and trained, varying in which augmentations were applied) (columns), through the projection head layers (bars). Y-axis is the accuracy of the decoder on a validation set requiring generalization of the decoded feature (e.g. color) to held-out values of non-target features (e.g. scales and positions).

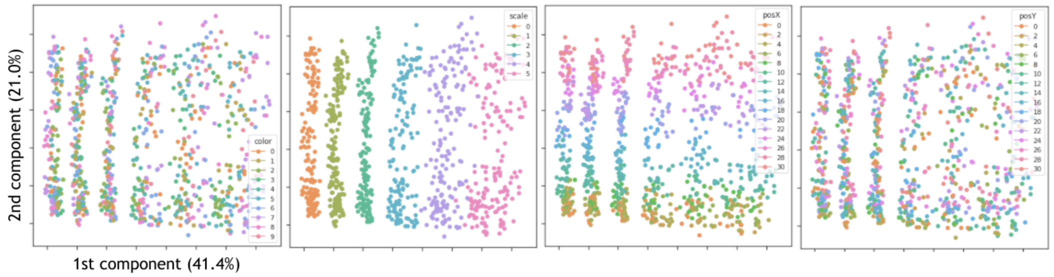

Figure 4: Organization by continuous features is apparent in principal components of the projection head input. PCA applied of activations for 1000 images, from a crop + jitter model. Each panel shows the embeddings colored by a different feature of the input image (left to right: color, scale, x-position, y-position).

head than at the output for a model trained with random crops and color jitter (Figure 3). We considered, as a baseline, a model without augmentation (a choice which results in an objective that merely pushes items apart). Qualitatively, we found similar patterns of decodability through the projection head as for the crop + jitter model, suggesting that the advantage of using the projection head input rather than the output for downstream tasks is not solely a consequence of enforcing invariance to augmentation at the projection head output as suggested by Chen *et al.* [7]. However, in general, using data augmentation comprising both random crops and color jitter enhanced feature decodability prior to the output. Training with color-incongruent random crop augmentation alone drove up color decodability in all but the final projection head layer. By contrast, color-congruent color jitter augmentation washed out color information but enhanced position decodability. Consistent with the decoding results, latent space visualization exposed organization according to appearance features of the input image (example shown in Figures 4 and B.5). These experiments show that both the importance of data augmentation for contrastive learning and the properties of SimCLR's projection head can be studied using simple synthetic data.

4 Discussion

Having shown that many of the seemingly puzzling behaviors of contrastive learning methods can be replicated and investigated on much simpler datasets than those used to validate their performance, we address two questions that these datasets are designed to support. First, why does a contrastive model learn features that are class-relevant? Using a dataset with discrete structure, we find that a gradient-based contrastive model naturally learns hierarchy, just as supervised models have been shown to do. Second, what does a contrastive model learn about continuously varying features, like color and size, shared across images? We replicate the utility of data augmentation and isolate the effects of augmentations on the representation of specific features through the network.

Acknowledgments and Disclosure of Funding

We thank Rafael Müller for helpful feedback.

References

[1] Bachman, P., Hjelm, R. D., and Buchwalter, W. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems* (2019), pp. 15535–15545.

[2] Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence 35*, 8 (2013), 1798–1828.

[3] Bilal, A., Jourabloo, A., Ye, M., Liu, X., and Ren, L. Do convolutional neural networks learn class hierarchy? *IEEE transactions on visualization and computer graphics 24*, 1 (2017), 152–162.

[4] Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., and Wanderman-Milne, S. JAX: composable transformations of Python+NumPy programs, 2018.

[5] Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599* (2018).

[6] Chen, R. T., Li, X., Grosse, R. B., and Duvenaud, D. K. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems* (2018), pp. 2610–2620.

[7] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709* (2020).

[8] Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029* (2020).

[9] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems* (2016), pp. 2172–2180.

[10] Eastwood, C., and Williams, C. K. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations* (2018).

[11] `https://github.com/google/flax/blob/master/examples/imagenet/resnet_v1.py`.

[12] He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 9729–9738.

[13] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778.

[14] Hénaff, O. J., Srinivas, A., De Fauw, J., Razavi, A., Doersch, C., Eslami, S., and Oord, A. v. d. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272* (2019).

[15] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework.

[16] Hinton, G. E. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society* (1986), vol. 1, Amherst, MA, p. 12.

[17] Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670* (2018).

[18] Kim, H., and Mnih, A. Disentangling by factorising. *arXiv preprint arXiv:1802.05983* (2018).

[19] Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning* (2019), pp. 4114–4124.

[20] Matthey, L., Higgins, I., Hassabis, D., and Lerchner, A. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

[21] Misra, I., and Maaten, L. v. d. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 6707–6717.

[22] Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[23] Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., and Tucker, G. On variational bounds of mutual information. In *International Conference on Machine Learning* (2019), pp. 5171–5180.

[24] Ridgeway, K., and Mozer, M. C. Learning deep disentangled embeddings with the f-statistic loss. In *Advances in Neural Information Processing Systems* (2018), pp. 185–194.

[25] Rogers, T. T., and McClelland, J. L. *Semantic cognition: A parallel distributed processing approach*. MIT press, 2004.

[26] Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120* (2013).

[27] Saxe, A. M., McClelland, J. L., and Ganguli, S. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences 116*, 23 (2019), 11537–11546.

[28] Saxe, A. M., McClellans, J. L., and Ganguli, S. Learning hierarchical categories in deep neural networks. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (2013), vol. 35.

[29] `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html`.

[30] `https://github.com/google-research/simclr`.

[31] Sohn, K. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems* (2016), pp. 1857–1865.

[32] Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849* (2019).

[33] Tschannen, M., Bachem, O., and Lucic, M. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069* (2018).

[34] Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 3733–3742.

[35] Ye, M., Zhang, X., Yuen, P. C., and Chang, S.-F. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE Conference on computer vision and pattern recognition* (2019), pp. 6210–6219.

Supplementary Material for “Understanding self-supervised learning using controlled datasets with known structure”

A Supplementary Methods

A.1 Hierarchical structure experiments

Model. *Architecture.* The model architecture consisted of a LeNet-style backbone and a 3-layer projection head. CNN backbone: 2 conv-ReLU-pool layers (conv layers: num_filters = [6, 16], filter_sizes = [5, 5], strides = [1, 1]; pool layers: 2×2 max pool with stride = 1). Projection head: 3 fully-connected layers with 120, 84, and 84 units, respectively, and ReLU’s after the first and second layers. It was implemented in JAX [4].

Hierarchically constrained similarity analysis. In Figures B.2 and B.3, to compare the within- versus across-class similarities of item representations across levels of the hierarchy, we constructed two probe sets, consisting of two views each of 200 items randomly sampled from the train and validation sets, respectively. The validation set consisted of randomly held out items (10% of the dataset). At a given level of hierarchy, within-class similarity was the mean of the pairwise cosine similarities between items belonging to the same class. Across-class similarity was the mean pairwise similarity between items belonging to different classes with the same parent level (hierarchically constrained, e.g. in a naturalistic setting, this might correspond to canaries vs. robins but not canaries vs. oaks).

A.2 Continuous appearance parameter experiments

Model. Each model was trained on the SimCLR objective [7] for 200 epochs, using a batch size of 128, an Adam optimizer with learning rate of 0.001, and a contrastive loss temperature of 0.5. We held the dataset and random network initialization fixed across experiments.

Architecture. We used a modified ResNet-18 [13] backbone (first convolutional layer: filter_size = 3, stride = 1, no pool) (modified implementation from [11]) with a 3-layer projection head (fully connected layers with units = [512, 512, 128]; the first two included a ReLU activation function, and the output did not include a bias).

Augmentations. Color jitter was a composition of color distortion (applied with probability = 0.8 and jitter strength = 0.5) and color dropping (with probability = 0.2) as implemented in [7, 30]. Random resized crops involved sampling uniformly at random a crop area ([8%, 100%] of the original image size) and aspect ratio ([0.75, 1.33] of the original aspect ratio) as described in [7] and implemented in [30].

Decoding. We trained decoders for 500 epochs, using a batch size of 256 and an Adam optimizer with a cosine learning rate decay schedule, and took the best validation accuracy over the training period over hyperparameter combinations (learning rates = [1e-2, 1e-3, 1e-4] × weight decays = [0, 1e-4, 1e-2]).

Latent space visualization. We performed PCA (implementation from [29]) on layer activations (including ReLU) elicited by 1000 randomly sampled train set images.

B Supplementary Figures

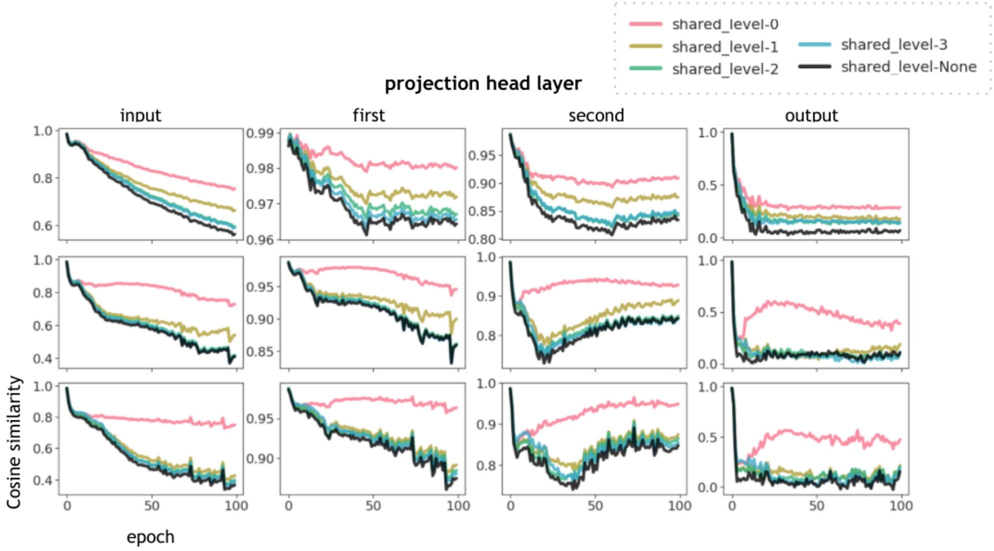

Figure B.1: Cosine similarity of reference items to probes where channels are populated with randomly drawn masks, rather than with masks sampled from the train set as in Figure 2. The results are qualitatively similar.

.

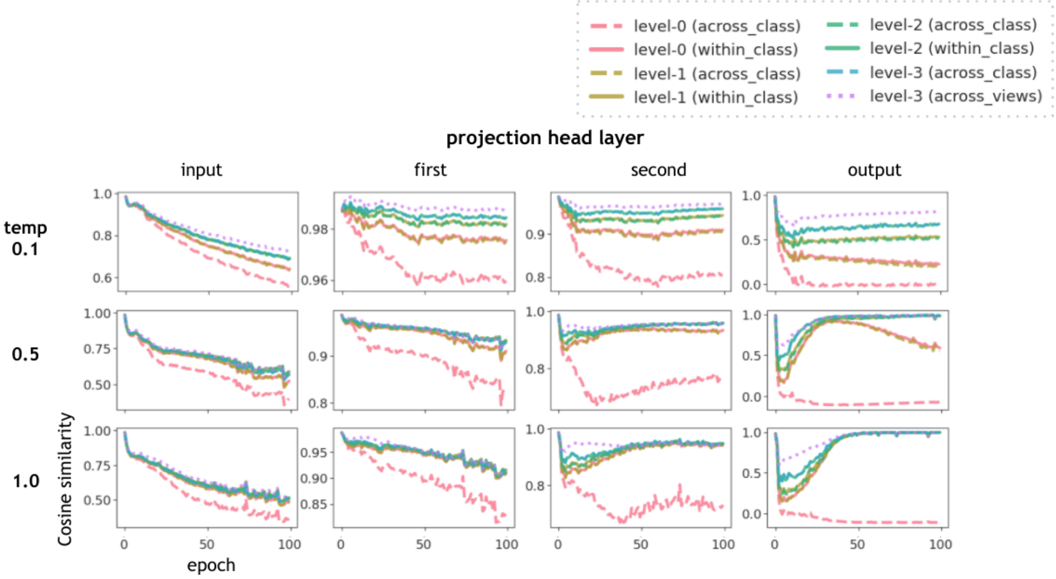

Figure B.2: Hierarchically-constrained similarity analysis. **SimCLR models learn item embeddings such that train set items belonging to different top-level classes exhibit the lowest cosine similarity (are well separated)**. Dashed lines indicate across-class, and solid lines indicate within-class, similarity at each level of the hierarchy. Dotted purple line indicates across-view (within-item) similarity. With a lower contrastive loss temperature (top row), SimCLR learns additional hierarchical structure, with similarity falling off as a function of hierarchy level. See Appendix A.1 for further details of this analysis.

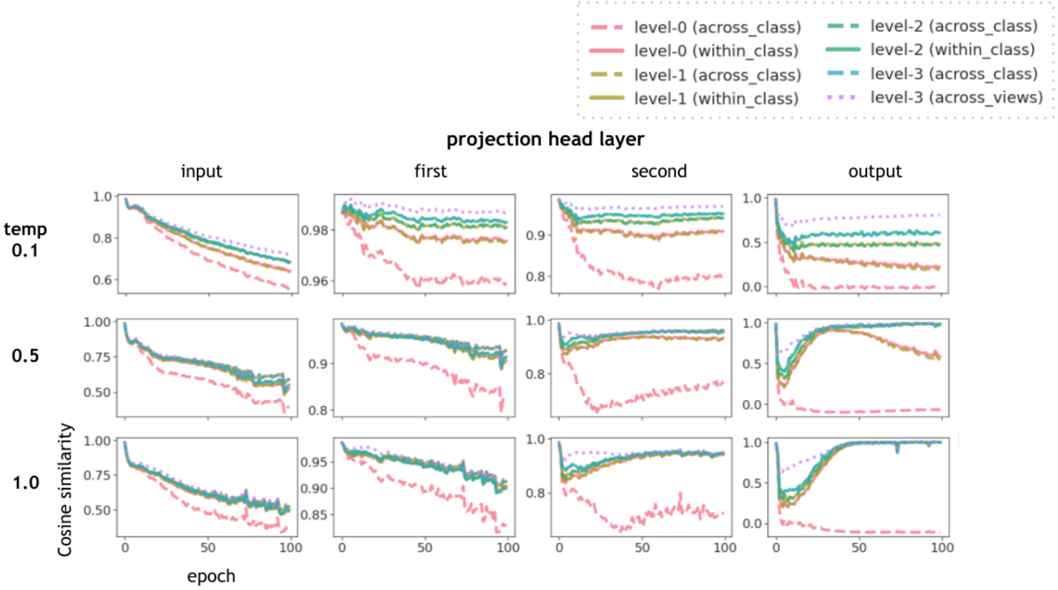

Figure B.3: The results of a hierarchically-constrained similarity analysis of items drawn from the validation set are highly similar to when items are drawn from the train set (Figure B.2).

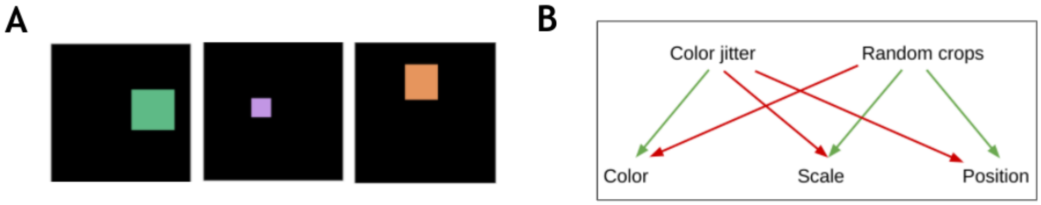

Figure B.4: **(A)** Example items from the modified `dSprites` dataset [20, 19]. Objects varied in color, scale, and position. **(B)** Congruent (green) versus incongruent (red) augmentation–feature pairs.

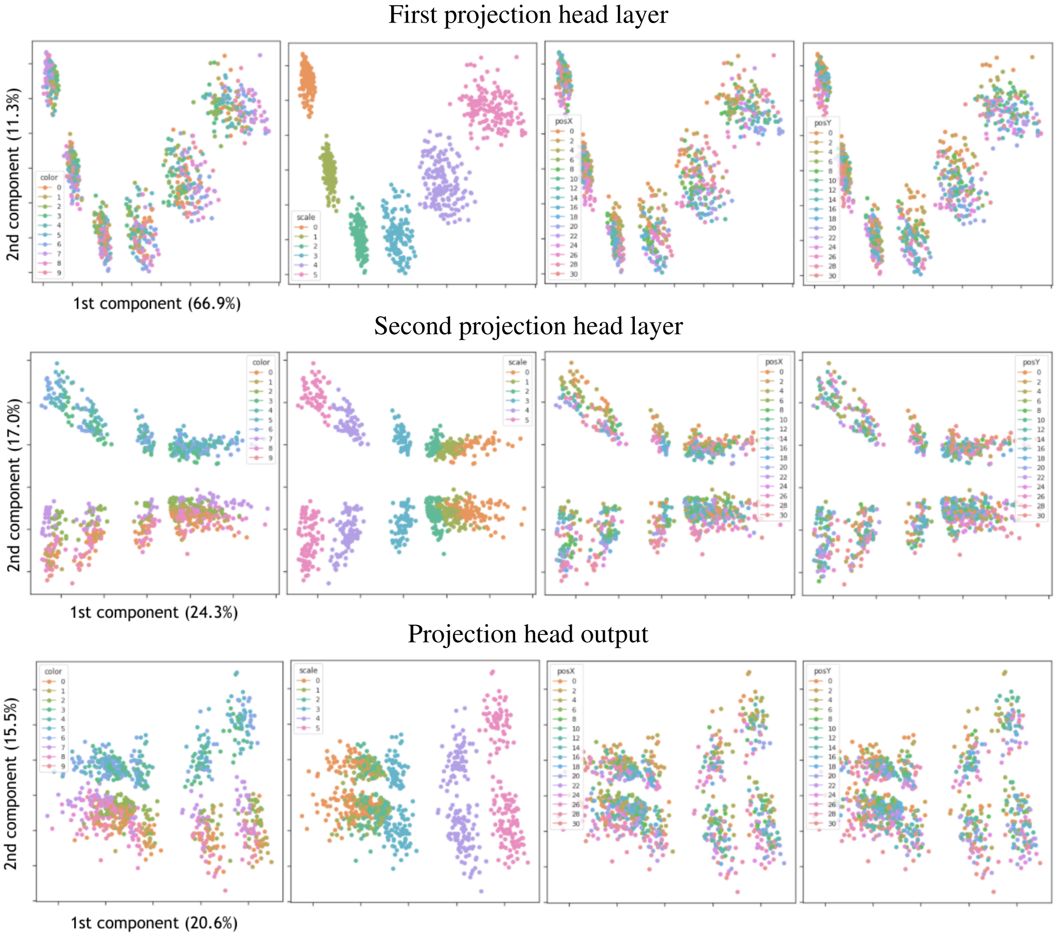

Figure B.5: Visualization of latent space of the color + jitter model, colored by (left to right) color, scale, x-position, y-position. See Figure 4 for projection head input.