STUDYING PHASE TRANSITIONS IN CONTRASTIVE LEARNING WITH PHYSICS-INSPIRED DATASETS

Ali Cy[1 *], **Anugrah Chemparathy**[1 *], **Michael Han**[1], **Rumen Dangovski**[1],
Peter Lu[1,2], **Marin Soljacic**[1]
[1]Department of Physics, Massachusetts Institute of Technology
[2]University of Chicago
{califyn, anugrah, mjhan, rdangovs, soljacic}@mit.edu,
lup@uchicago.edu

ABSTRACT

In recent years contrastive learning has become a state-of-the-art technique in representation learning, but the exact mechanisms by which it trains are not well understood. By focusing on physics-inspired datasets with low intrinsic dimensionality, we are able to visualize and study contrastive training procedures in better resolution. We empirically study the geometric development of contrastively learned embeddings, discovering *phase transitions* between locally metastable embedding conformations towards an optimal structure. Ultimately we show a strong experimental link between stronger augmentations and decreased training time for contrastively learning more geometrically meaningful representations. Our code is available here.

1 INTRODUCTION

One of the key problems in modern machine learning is crafting effective representations of data without human-generated labels. Contrastive learning and self-supervised learning methods are among the most popular and effective methods to date for tackling this problem. In practice, self-supervised learning methods require a large output dimensionality for optimal performance (Chen et al., 2020; Chen & He, 2021; Grill et al., 2020; Bardes et al., 2022).

There is still little insight into the mechanisms by which contrastive learning works. Some prior work has been done in understanding the contrastive training process through augmentation graphs (HaoChen et al., 2021; Wang et al., 2022). From a theoretical perspective, Wang & Isola (2020) and Zimmermann et al. (2021) proved that the optimal contrastive representation is uniform and a linear transformation of the latent space under some conditions.

Visualization is an exceptionally powerful tool for analyzing learned representations, but is difficult because of the typical large dimensionality of representations. Dimensionality reduction and/or aggregation is necessary, which destroys much meaningful information, such as in (Zhu et al., 2021). Contrastive learning can also create interpretable low-dimensional representations (Hua et al., 2021). However, because the intrinsic dimensionality of datasets such as CIFAR-10 is quite high, the real embeddings in high-dimensional space may look radically different.

Instead, we propose visualizing learned representations of a dataset that has a low intrinsic dimensionality but rich structure. In order to do this, we use data generated from a classic physics problem: extracting the three conserved quantities from instantaneous observations of position and velocity of a gravitational orbit. Using "Kepler" datasets, we are able to visualize and interpret the changing output geometry of contrastively trained toy networks. Ultimately we discover a replicable phase transition between two output geometries and show the importance of augmentation strength in accomplishing the phase transition earlier in training.

* denotes equal contribution

1.1 Related work

Augmentation strength and quality of representations. Commonly, it is understood that there is a "sweet spot" in terms of augmentation strength for which the amount of mutual information contained in the resultant embeddings is optimal for performance on a downstream task (Tian et al., 2020). In particular, augmentation overlap between similar samples is important in contrastive learning (Wang et al., 2022).

Contrastive learning on low-dimensional systems. Previous work has been done on contrastive learning on datasets with low intrinsic dimensionality. In particular the 3DIdent dataset (Zimmermann et al., 2021) uses orientation, position, and color of an object to parameterize the latent space. In these datasets, positive pairs are created by sampling the generated output corresponding to the nearest neighbors in the latent space.

Our Kepler dataset is equally simple to study but has a more natural temporal augmentation. Moreover, the dynamics of image generation or ConvNet training can be controlled for in our study, and our results are more applicable to contrastive learning in non-computer vision domains.

Deep learning to uncover conserved quantities. Prior work on discovering conserved quantities from the geometry of trajectory data used a variety of approaches, including manifold learning combined with symbolic regression (Liu & Tegmark, 2021), manifold identification in a known symplectic geometry (Mototake, 2021), manifold learning with an optimal transport metric (Lu et al., 2022), and regression with a Siamese network (Wetzel et al., 2020). However, we use our physics dataset to analyze training dynamics and do not focus on the performance of our method.

2 Preliminaries

2.1 Contrastive learning

We use the SimCLR framework outlined by Chen et al. (2020) with variants of the InfoNCE loss (Oord et al., 2018).

Consider an encoder network $f(x)$ and a batch of b inputs, $\{(x_i^1, x_i^2)\}_{1\le i\le b}$, where x_i^1 and x_i^2 are two randomly augmented versions of the same data, which together form a positive sample. Then the contrastive loss can be written as

$$\ell = -\mathbb{E}_i \left[\log \frac{e^{-\delta(f(x_i^1), f(x_i^2))}}{e^{-\delta(f(x_i^1), f(x_i^2))} + \sum_{j\neq i} e^{-\delta(f(x_i^1), f(x_j^2))}} \right],$$

where δ is a distance function. We use two variants of the InfoNCE loss based on different δ functions: the traditional cosine distance (termed CosNCE) and Euclidean distance (termed L2NCE). For the Kepler experiments, we use L2NCE to facilitate visualization (Böhm et al., 2022), whereas for CIFAR-10 experiments, we use standard CosNCE. More details can be found in Appendix A.

2.2 Kepler Orbits Datasets

The elliptical orbit of a planet around a central star can be uniquely defined by three conserved scalar quantities: energy H, angular momentum $\|\mathbf{L}\|$, and the angle of the Laplace-Runge-Lenz vector ϕ_0.

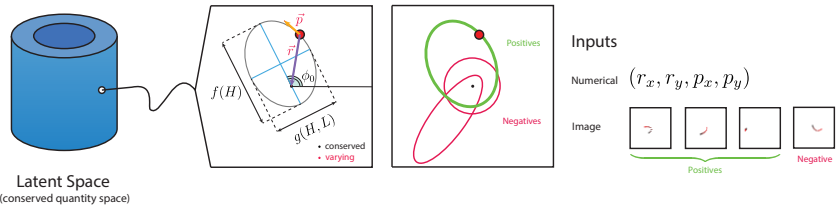

Figure 1: Data generation for our Kepler dataset with example numerical and image data.

These conserved quantities can be reconstructed from position $\mathbf{r}$ and momentum $\mathbf{p}$ with the above where $\arg(\mathbf{v})$ is the angle of vector $\mathbf{v}$ with respect to the positive x-axis.

$$H = \frac{\|\mathbf{p}\|^2}{2} - \frac{1}{\|\mathbf{r}\|} \qquad \mathbf{L} = \mathbf{r} \times \mathbf{p} \qquad \phi_0 = \arg(\mathbf{p} \times \mathbf{L} - \hat{\mathbf{r}})$$

In the numerical domain, we use different states (i.e. position-momentum pairs, a 4-vector) of the same orbit at different times to produce positive examples for learning representations of the orbits, and states from different orbits as negative examples. Details for the image domain are in Appendix A.1. Our time-based augmentation is similar to sampling from video data (Dave et al., 2022; Qian et al., 2021) in encouraging the representations to become temporally invariant. We use datasets of 10,240 newly generated orbits, each with 10 time samples.

The latent space for generating orbits from these conserved quantities is three-dimensional, so the output embeddings can be visualized in 3 dimensions directly. The exact intervals from which we sampled H and L, as well as that choice's impact on representation geometry, are in Appendix B.

Partial trajectories. We can weaken our time augmentations by restricting the range of times for positive examples. Formally, set α to be the proportion of the orbit that positive samples are sampled from. Let T be the period of the orbit. For each orbit, we uniformly sample a starting time t from $[0, T]$, and then we sample positive examples uniformly from the range $[t, t + \alpha T]$.

3 Analyzing Training Dynamics through Visualization

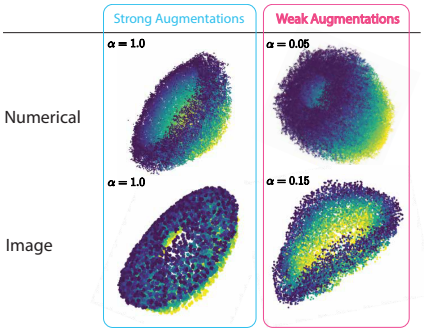

Figure 2: Representation geometry for models trained on numerical and image Kepler data with full and partial trajectories, colored by angular momentum L.

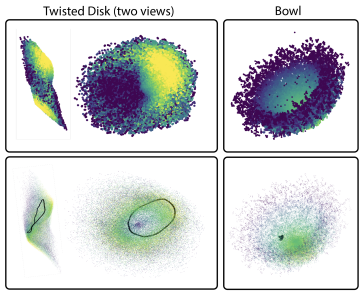

Figure 3: Visual comparison showing poor alignment in Twisted Disk, but not in the Bowl. Black outline is embedding of a single orbit.

The Kepler orbit problem is easily solved using our contrastive framework in both the numerical and image domains. All three conserved quantities are clearly present in the representation, and it showcases the geometry of the latent space in nontrivial ways; see Figure 2 and Appendix B. We have nicknamed this particular representation the *Bowl* because of its shape. We can also visualize the development of the representation geometry towards the Bowl shape during training (Appendix C).

When weakening the augmentation strength α, the bowl can still be recovered rather easily (Figure 2). Although global structure is generally preserved, alignment degrades considerably.

As α approaches zero, the contrastively trained network begins to see less and less of the possible augmentation space of its input dataset. This intuitively forces the model to optimize the geometries of local clusters in the embedding space instead of being able to make large geometric reorganizations in a small handful of epochs.

Through the Kepler datasets we present visualizations of global *phase transitions* with only local deformations. The loss does not sharply drop during the phase transition (Figure 18) and as a result the transitions can only be found through visualization.

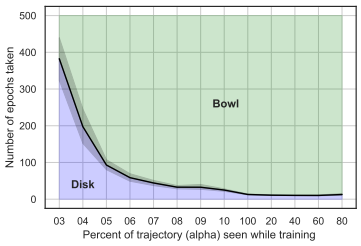

Figure 4: Number of epochs needed to reach and complete final unwrapping of the bowl phase transition for Kepler charted against percentage of trajectory seen during training. 16 trials for each α, with a 95% confidence interval (in black).

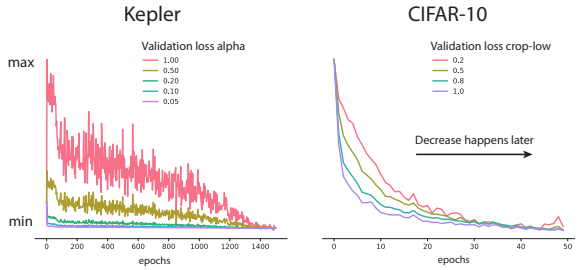

Figure 5: Tracking multiple validation losses, i.e. normal contrastive loss but calculated on datasets with increasing augmentation strength, on Kepler and CIFAR-10 models trained with weak augmentations ($\alpha = 0.05$, croplow 0.8, respectively). Each loss curve is normalized so that the top of the graph is maximum and the bottom is minimum. See Appendix H for controls.

3.1 Observing Embedding Evolution from "Twisted Disk" to "Bowl"

In the numerical domain, when we set $\alpha \leq 0.01$, the shape of the representations looks radically different. The representations remain stuck at a shape that we have named the *Twisted Disk* (Figure 3), which has a twist at the center at which orbits with $L \approx 0, 1$ are embedded together. Despite being able to achieve low loss with partial trajectories (Figure 5), the disk is misleading about the global structure of the data. Visually the orbits also align poorly, maintaining their elliptical shape - the disk is effectively a direct embedding of the position/momentum input space.

In contrastive learning, models have been observed to take "shortcut" solutions or suppress features (Chen et al., 2021; Robinson et al., 2021). The disk geometry is an good toy model of this phenomenon: as the strength of the time augmentation decreases, the model begins to embed time-variant quantities, especially position.

We observed that the timing of phase transitions (tracked quantitatively through a regression metric on the representation geometry - Appendix G) seems to be controlled by the augmentation strength (Figure 4). This suggests a continuous element to whether or not certain phase transitions can occur; instead of either occurring or not occurring, they take longer and longer to occur as augmentations are weakened. We observe the same phenomenon in the image domain, except with multiple phases and more complicated training dynamics. A detailed treatment can be found in Appendix F.

3.2 Global structure emerges gradually during training

The reason why weaker augmentations prolong the time needed for phase transitions is unclear. We stress a distinction between *local* and *global* structure in representations, with respect to distances in the embedding space. Models gradually recover global structure by piecing together local chunks of information. Intuitively, as augmentation strength decreases, each chunk gets smaller, so more training iterations are needed for global structure to be recovered. Therefore, any decrease on a validation loss computed with respect to a dataset with stronger augmentations—which requires more global structure—should happen *after* similar decreases on a validation loss with weak augmentations.

To test this, we monitored multiple validation losses, evaluated on datasets containing a series of increasingly strong augmentations, during training of both Kepler and CIFAR-10 models (Figure 5). More detailed experimental results are in Appendix H.

4 Conclusion

Through visualizations on a low-dimensional physics dataset, we have shown the existence of phase transitions in contrastive learning. We propose a more subtle understanding of the effect of weak augmentations, wherein they slow down training by encouraging contrastive models to focus on local structure.

REFERENCES

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2105.04906.

Jan Niklas Böhm, Philipp Berens, and Dmitry Kobak. Unsupervised visualization of image datasets using contrastive learning, 2022. URL https://arxiv.org/abs/2210.09879.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/2002.05709.

Ting Chen, Calvin Luo, and Lala Li. Intriguing properties of contrastive losses. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=rYhBGWYm6AU.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *Conference on Computer Vision and Pattern Recognition*, 2021. URL https://arxiv.org/abs/2011.10566.

Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljacic. Equivariant contrastive learning. *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2111.00899.

Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. TCLR: Temporal contrastive learning for video representation. *Computer Vision and Image Understanding*, 219: 103406, jun 2022. doi: 10.1016/j.cviu.2022.103406. URL https://doi.org/10.1016%2Fj.cviu.2022.103406.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *CoRR*, abs/2006.07733, 2020. URL https://arxiv.org/abs/2006.07733.

Jeff Z. HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Neural Information Processing Systems*, 2021. URL https://arxiv.org/abs/2106.04156.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *Conference on Computer Vision and Pattern Recognition*, 2020. URL http://arxiv.org/abs/1911.05722.

Tianyu Hua, Wenxiao Wang, Zihui Xue, Yue Wang, Sucheng Ren, and Hang Zhao. On feature decorrelation in self-supervised learning. *International Conference on Computer Vision*, 2021. URL https://arxiv.org/abs/2105.00470.

Ziming Liu and Max Tegmark. Machine learning conservation laws from trajectories. *Phys. Rev. Lett.*, 126:180604, May 2021. doi: 10.1103/PhysRevLett.126.180604. URL https://link.aps.org/doi/10.1103/PhysRevLett.126.180604.

Peter Y. Lu, Rumen Dangovski, and Marin Soljačić. Discovering conservation laws using optimal transport and manifold learning, 2022. URL https://arxiv.org/abs/2208.14995.

Yoh-ichi Mototake. Interpretable conservation law estimation by deriving the symmetries of dynamics from trained deep neural networks. *Phys. Rev. E*, 103:033303, Mar 2021. doi: 10.1103/PhysRevE.103.033303. URL https://link.aps.org/doi/10.1103/PhysRevE.103.033303.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018. URL https://arxiv.org/abs/1807.03748.

Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6964–6974, June 2021.

Joshua David Robinson, Li Sun, Ke Yu, kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. Can contrastive learning avoid shortcut solutions? In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=ud-WYSo9JSL.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2005.10243.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/2005.10242.

Yifei Wang, Qi Zhang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Chaos is a ladder: A new theoretical understanding of contrastive learning via augmentation overlap. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=ECvgmYVyeUz.

Sebastian J. Wetzel, Roger G. Melko, Joseph Scott, Maysum Panju, and Vijay Ganesh. Discovering symmetry invariants and conserved quantities by interpreting siamese neural networks. *Phys. Rev. Res.*, 2:033499, Sep 2020. doi: 10.1103/PhysRevResearch.2.033499. URL https://link.aps.org/doi/10.1103/PhysRevResearch.2.033499.

Rui Zhu, Bingchen Zhao, Jingen Liu, Zhenglong Sun, and Chang Wen Chen. Improving contrastive learning by visualizing feature transformation. *International Conference on Computer Vision*, 2021. URL https://arxiv.org/abs/2108.02982.

Roland S. Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. *International Conference on Machine Learning*, 2021. URL https://arxiv.org/abs/2102.08850.

A MORE TRAINING DETAILS

Our model architectures are as follows:

- For numerical experiments on the Kepler dataset, we use ReLU MLP networks with four hidden layers of width 64 trained with the L2NCE loss.
- For image experiments on the Kepler dataset, we use a ResNet-18 backbone, with its final fully-connected layer replaced by the same ReLU MLP network used for numerical experiments. The model is trained with the L2NCE loss.
- For experiments on CIFAR-10, we use a ResNet-18 encoder with a one-hidden layer projector with hidden size 2048 and output size 2048. We use a CIFAR version where the first layer has kernel size 3, stride 1, and padding 1. The model is trained with the CosNCE loss and all standard hyperparameter choices, following (Dangovski et al., 2022).

Notably we do not use a projector in our Kepler dataset experiments in order to force encoder representations to be time invariant, allowing for better visualization of the geometric structure of the latent space.

A.1 IMAGE DATASET GENERATION

We generate simple, 56×56-dimensional images based on the Kepler dataset. Because both position and velocity are needed to encode the three conserved quantities, we start with a blank image and fill in a sequence of ten positions along the orbit, spaced 0.1 time units apart with a gradient from black to red, giving the visual appearance of a comet tail. While the images are relatively simple, this dataset has a relatively high dimensionality like that of real images while keeping the exact same structure as the numerical version.

B DISCUSSION OF REPRESENTATION GEOMETRY

B.1 RELU NETWORKS: THE BOWL GEOMETRY

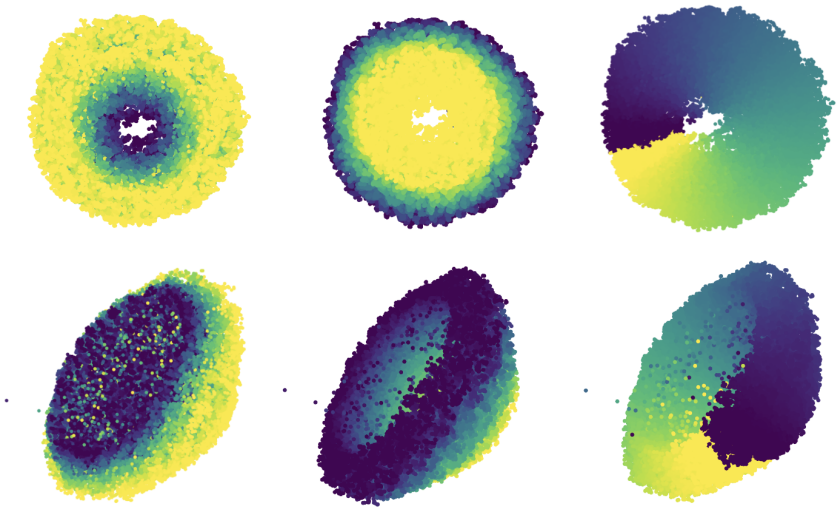

Figure 6: Bowl representation geometry obtained from a ReLU network trained over 1500 epochs with L2NCE loss presented from two different angles and colored by conserved quantities H, L, ϕ_0 from left to right.

The bowl nontrivially represents the latent geometry in the following ways:

1. Orientation ϕ_0, which is a rotational component, is encoded rotationally, so that $\phi_0 = 0, 2\pi$ are encoded to be at the same location. H, L are encoded linearly.
2. The bowl has a hollow interior, which makes it a *bowl* as opposed to a *cone* (see B.5). This is because of the data generation process we use; for any given momentum L, the range of

possible energies H cuts off at or above the true physical minimum, meaning that parts of the dataset are "missing".

3. The rotational axis of symmetry, about which we can define orientation ϕ_0 (and on which ϕ_0 is ill-defined), corresponds to the lowest given H for any given L, which is also when the orbit is perfectly circular and thus its orientation ϕ_0 becomes ill-defined.

We can validate this visually by plotting slices of the input space (as below), or empirically by polynomial regression of degree 2.

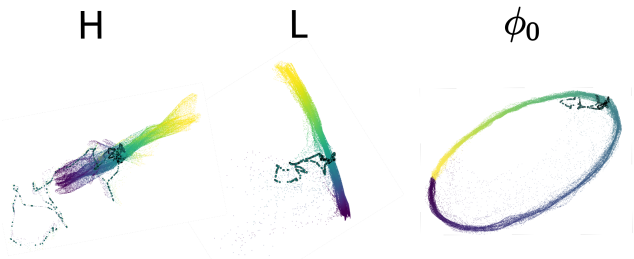

Figure 7: Portion of overall representation geometry corresponding to fixing two conserved quantities and varying the other (i.e. $H \in [-0.5, -0.25], L = 0.5, \phi_0 = \pi$ in the leftmost embedding) colored by the remaining conserved quantity. An additional single orbit (all quantities fixed - i.e. $H = -0.375, L = 0.5, \phi_0 = \pi$ in the leftmost embedding) is also plotted as denser, shaded circles.

B.1.1 Figure-Eight Bowl and other rare variants

Occasionally, when training with short trajectories, we observe a *figure-eight* variant of the bowl, which is identical to the bowl but ϕ_0 is encoded rotationally around a figure-eight instead of a circle (Figure 8). It appears to be an artifact of the training data; it happens particularly often (20-50%) for some datasets generated with the same general parameters.

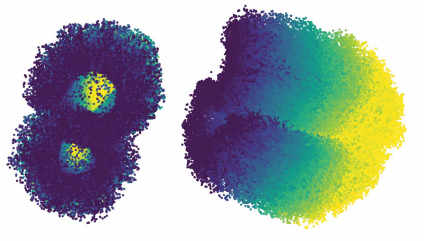

Figure 8: Two views of figure eight bowl, colored by L.

More rarely, we have observed other bowl variants, e.g. ones with an extremely thin H axis, or flattened into 2D (dimensional collapse). These, similarly, appear to be hinged on unusual generations of the input data. They converge for very long training times to various bowl variations (> 1000 epochs) but will "retrain" into the normal bowl if trained on other random generations of the same input data. Additionally, shapes tend to change slightly when changing training parameters, e.g. BatchNorm, although the difference tends to be minimal in most cases.

B.2 Sigmoid Networks: more Twisted Disks

One rather important training parameter is the choice of activation function. We have already seen the subpar Twisted Disk geometry as the primary precursor to the Bowl geometry in subsection 3.1. It turns out that trying to train a sigmoid-activation network with L2NCE loss also leads to the twisted disk conformation (albeit with better alignment), and never makes it past this local minimum on the loss landscape. One difference between this Twisted Disk and the other is the hole in the middle.

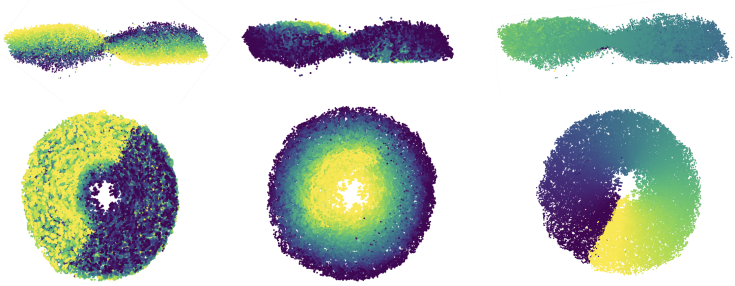

Figure 9: Twisted Disk representation geometry obtained from a sigmoid network trained over 1500 epochs with L2NCE loss presented from two different angles and colored by conserved quantities H, L, ϕ_0 from left to right.

Notably this variant of the twisted disk obtains significantly better alignment than the version showed in Figure 3, however empirically evaluating the contrastive loss shows the inferiority of this conformation. Plotting the training loss clearly shows the Bowl geometry (Figure 10) better optimizes the loss compared to the Twisted Disk geometry.

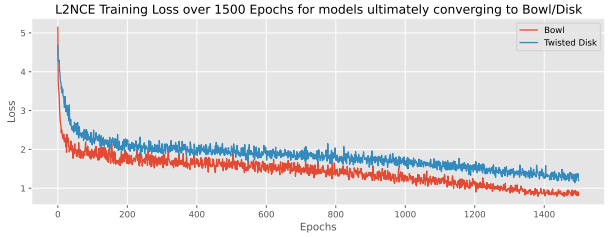

Figure 10: Twisted Disk representation geometry obtained from a sigmoid network trained over 1500 epochs with L2NCE loss presented from two different angles and colored by conserved quantities H, L, ϕ_0 from left to right.

Currently we hypothesize that this due to the smoothness of a random-initialized sigmoid networks deformations on input manifolds making it difficult for the network to slowly phase transition local pieces of the geometry into the Bowl shape.

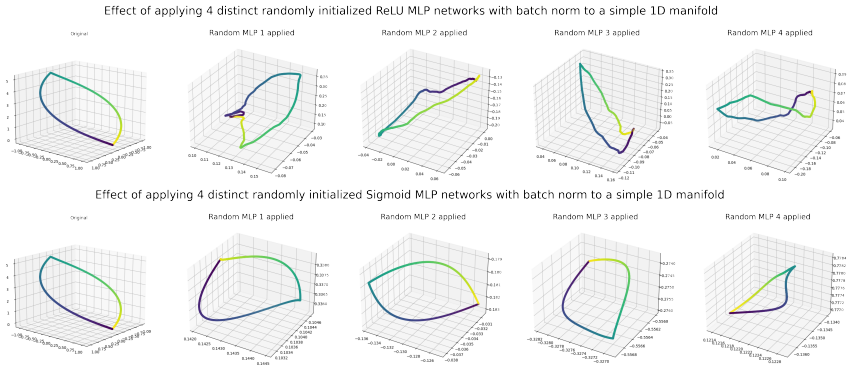

Figure 11: Toy demonstration comparing smoothness of output for randomly intialized ReLU v.s. Sigmoid networks on a simple loop.

B.3 Getting around the hypersphere: CosNCE and "traditional" SimCLR

We can follow SimCLR (Chen et al., 2020) more exactly by using CosNCE and a projector head. We attempt to bypass the normalization of the CosNCE loss chewing up a visualization dimension by using the nonlinear projector to "convert" a geometrically sensible output space from $\mathbb{R}^3$ to S^3.

We obtain our best results by using an encoder output dimension of 3 and a nonlinear 3-layer projector with output dimension of 4. Although the training of the CosNCE models by this approach is somewhat tricky, we ultimately are able to also extract the bowl representation with some level of distortion (Figure 12).

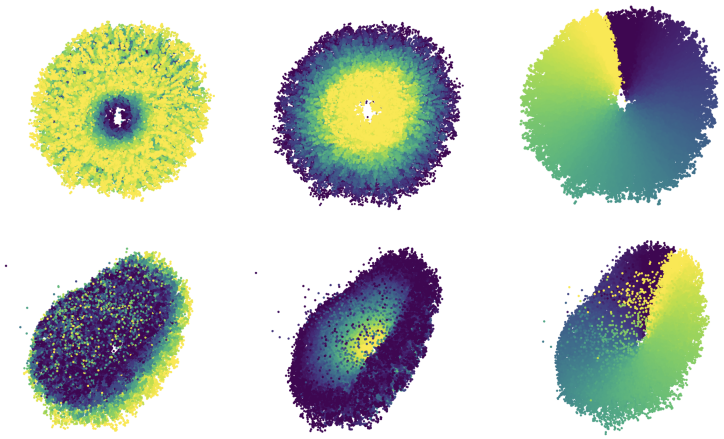

Figure 12: Bowl representation geometry obtained from a ReLU encoder (with projector discarded) trained over 1500 epochs with CosNCE loss presented from two different angles and colored by conserved quantities H, L, ϕ_0 from left to right.

Because traditional cosine similarity InfoNCE provided an clear conceptual framework for contrastive learning, we did not try MoCo (He et al., 2020), but since it is also a contrastive method, we expect it would likely have similar behavior to SimCLR as opposed to other methods.

B.4 Non-contrastive methods

Generally, recovering all three conserved quantities was much more difficult with non-contrastive methods, which was one of the reasons we did not study them in detail. We did recover a bowl shape for VICreg (Bardes et al., 2022), but were not able to with SimSiam (Chen & He, 2021).

B.5 Complete input space

We picked gravitational parameter $\mu = Gm = 1$. We chose to restrict $H \in [-0.5, -0.25]$ and $L \in [0, 1]$ to provide numerical stability for the training. In reality, $H \in [-\infty, 0], L \in [0, \infty]$ are okay as long as the eccentricity $\sqrt{1 + \frac{2HL^2}{\mu^2}}$ is between 0 and 1. We can get a closer approximation of the true input space by allowing H, L to vary on larger intervals given that the eccentricity condition is not violated. When trained on such a dataset, we get a cone shape:

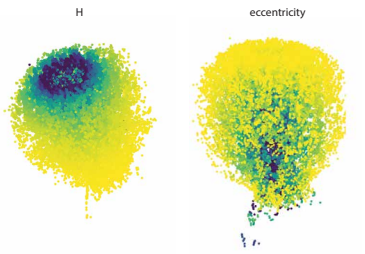

Figure 13: Model trained on complete Kepler dataset visualizing the complete Kepler dataset: "cone". Eccentricity is used instead of L because it is more easy to interpret; it is also a "conserved quantity".

This requires setting a much lower learning rate (0.005 vs 0.05 normally).

C VISUALIZING TRAINING DYNAMICS: MANIFOLD DEFORMATION IN RADIAL BOWL COMPONENT

Thanks to the low dimensional output space, we can visualize individual pieces of a model's output representation geometry being deformed into the their final shape. Here, a two-dimensional input manifold (sampled from a 1D slice of the 3D latent space), parameterizable by (H, t) (energy and time sampled from orbit, with L, ϕ_0 fixed), folds into a line segment-like shape along which the main axis of variation is along H, whereas it is mostly invariant to t.

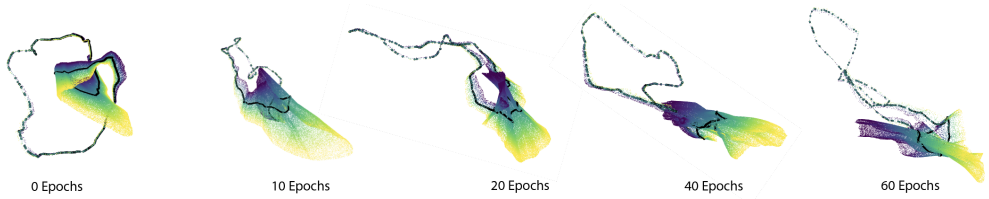

Figure 14: Changing embedding space over training time for 10,240 datapoints of varying H with constant L, ϕ_0, with a single orbit also plotted in denser, shaded circles (as in Figure 7).

There are two important features to address:

- The untrained network in Figure 14 produces a *naively* reasonable result when colored by H because a randomly initialized network produces a simple deformation on the input manifold of orbits of varying H (see Figure 11). This phenomenon is not equivalent to "solving" local or global structure - we seek a less trivial transformation on the manifold (as is accomplished after 60 epochs).
- In these developing manifolds we see long, narrow *tails* which are not condensed well in the folded manifold. These correspond to a relatively narrow time-slice of the orbits of the same energy during which the orbit is near the origin and thus moving with high velocity. These points are seldom sampled in training, and are intrinsically hard to distinguish, leading to poor alignment compared to the rest of the manifold. These "tail" points can be seen on the interior of the H-colored bowl in Figure 10.

We also see occasional instances of delayed local optimization. In small chunks of a Bowl representation, the locally line-segment like manifold unfolds and refolds to a new shape that does a slightly better job at condensing the orbit tails, and results in a long term decrease in loss (corresponding to movement to a slightly better local minima in the loss landscape).

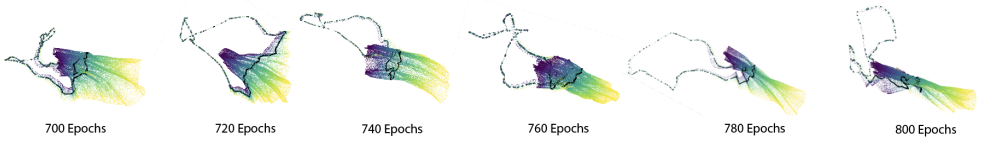

Figure 15: Delayed optimization in embedding space of the same (H, t) manifold as Figure 14 after 700 epochs. The formerly slightly crumpled manifold unfolds and reshapes over 100 further epochs allowing the geometry to better separate tails.

D OVERLY STRONG AUGMENTATIONS

The Kepler dataset intrinsically has perfectly meaningful augmentations. In order to create augmentations that are *too strong*, one thing we can do is to use the Kepler image dataset and restrict the image bounds so that occasionally planets will leave the boundary of the image and result in a blank, white image. Although this is not strictly an augmentation, the existence of a "nonsense" input (blank, white image) that is a positive example with many normal images is similar to the effect induced by too-strong augmentations where meaningful information gets destroyed and some augmented images lose any distinction that belonged to what they were originally. The existence of

such blank images tends to drastically change representation geometry, generally for the worse, by creating a central "point of attraction" that many points in the representation must be close to.

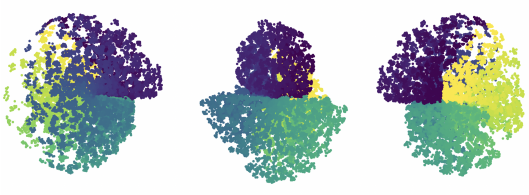

Figure 16: If we decrease the bounds of the image generation until a sizable (around 10%) of the generated images are blank, the bowl geometry gets heavily distorted as slices of the bowl attempt to approach the "nonsense" image center. Colored by H.

E Noise in Observations

For each orbit being generated, we can add Gaussian noise directly to each conserved quantity for each time step that is being evaluated. As the orbits are generally nonlinear in the conserved quantities, this prevents different orbits from experiencing effectively different levels of noise. This noise is equivalent to increasing the connectivity of the augmentation graph proposed by HaoChen et al. (2021) by introducing nearest-neighbor augmentations.

Typically in the contrastive learning setting, it is assumed that the latent variables between positive pairs are close but not identical (Wang & Isola, 2020). In our setting, we replicate this by adding a small amount of Gaussian noise to the conserved quantities. In the partial trajectory setting, this noise vastly shortens the time needed for the disk-bowl phase transition.

F Phase changes in the image domain

Generally, the bowl emerges later and later as α is decreased. Due to the increased difficulty of the image dataset, this generally happens with larger α than on the numerical dataset.

α	Epochs to bowl
0.25	by 60 epochs
0.15	by 250 epochs
0.10	by 1000 epochs

The Kepler image dataset invites considerably more complex training dynamics then the numerical dataset. Non-phase transition phenomena we observed were:

- Numerous stable variations of the bowl (as well as other representations), which different augmentation strengths ended up favoring differently. In particular, low α (less than 0.15) tend to first converge to a bowl and then two opposite sides draw close together, forming a taco-like shape.
- Medium α or specific data generations incentivized a bowl with one corner flipped over (i.e. for a range of ϕ_0, the H-axis (depth axis) was inverted), which was previously seen as a transitional form between disk and bowl in the numerical data, but we did not see a transition into a bowl before 4000 epochs.
- Backwards phase transitions: sometimes the model would reach a bowl, convert back into a disk-like shape, and then re-converge onto the bowl relatively quickly.
- Generally, more significant quality degradation even when the basic bowl shape was maintained.

G TRACKING PHASE TRANSITION TIMING WITH LINEAR REGRESSION

We are able to numerically track the last stage of the phase transition producing the bowl using linear regression on conserved quantities. Because (in the Kepler orbits dataset) the final representation geometry is roughly linear in angular momentum L, and the transitional forms/twisted disk show essentially no linear relationship in L (from visualization of hundreds of model checkpoint embeddings during testing - for example as in Figure 18) we used L for such evaluations.

This metric helps us to observe phase transitions, even though these large geometry changes do not correspond to large decreases in loss. Consider the following example run using a training dataset with $\alpha = 5\%$ visible partial trajectories. The loss steadily decreases, even beyond the proposed phase transition (65 - 85 epochs) but the regression metric effectively captures the phase transition into the bowl geometry.

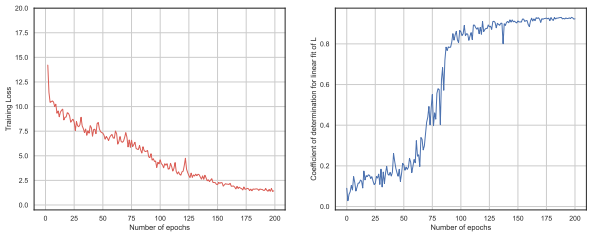

Figure 17: Comparing the training loss and linear regression metric for a standard ReLU network trained for 200 epochs. The phase transition (occurring between 65-85 epochs) visible through the regression metric is not apparent from the training loss.

We can also visualize the changing representation geometry during these epochs to observe the phase transition.

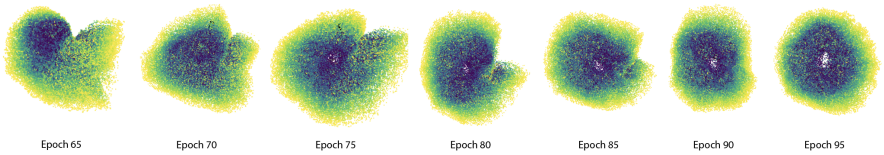

Figure 18: Representation geometry of the model during training between 65 - 95 epochs. Much of the phase transition occurs between 65-85 epochs - the remaining epochs resolve a kink in the bowl. Geometry colored by H to best show resolving deformation.

Repeating this metric-tracking for many trials allows us to roughly visualize the distribution of phase transition times using this procedure when $\alpha = 0.5$. The steep increase in R^2 proves to be an extremely good signal for the final stage of the phase transition when verified by visualization. We produce Figure 4 by tracking the time taken for the average R^2 of the past 3 epochs to reach 0.8 (as in the above image) for several different α.

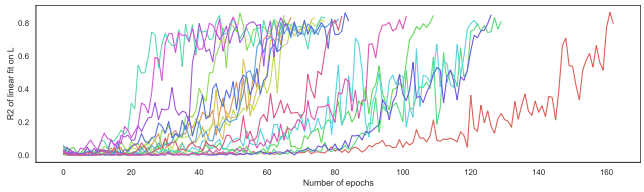

Figure 19: Tracking R^2 value for linear fit of network outputs on a randomly generated validation dataset of 1024 orbits on conserved quantity L during the training process over training with $\alpha = 0.5$. 16 fresh runs (training and validation datasets) were tracked and logged until they satisfied $\text{avg}(R^2_{t-3:t}) > 0.8$.

H MORE ON TRACKING MULTIPLE LOSSES

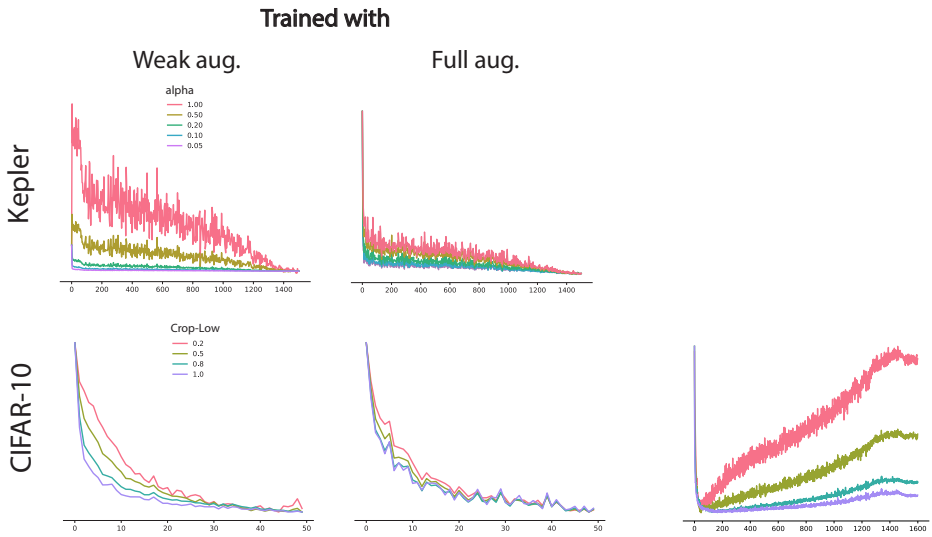

Figure 20: Tracking multiple losses with a full augmentation control, and also further on CIFAR-10 with weak augmentations, demonstrating overfitting.

The loss on strongly augmented datasets dropping later (relatively) is also present with full augmentations. However, this effect is much less prominent then on the weakly augmented version (Figure 20). The spike in the full augmentation model is due to collapse occurring at around 200 epochs and reversing at around 300 epochs in this particular instance; this is a rather rare occurence in our experience but may merit more investigation.

Additionally, past fifty epochs, the weakly augmented CIFAR-10 model (with crop low at 0.5) tends to overfit on all four losses, at which the delay effect no longer becomes interpretable because all four losses start to rise, which is why this figure cuts off at 50 epochs. No overfitting is present on the full augmentation model. Loss on stronger augmentations tends to rise the fastest. Less augmentation weakening may allow this delay effect to persist for longer before overfitting occurs. We also note that the variation *between* versions are substantial or even larger than variations in loss *within* a single model.

I CONNECTEDNESS IS NOT STRICTLY NECESSARY FOR GOOD REPRESENTATIONS

Contrastive learning has been analyzed through the lens of augmentation graphs, where an augmentation graph is a graph on the input support where there are edges between any two vertices correspoding to inputs sharing an augmentation. Oftentimes, the connectedness of such a graph within a class has been used in previous work as a theoretical assumption in contrastive learning (HaoChen et al., 2021; Wang et al., 2022), and it is often assumed in practice that such connectedness is necessary for classification.

However, contrastive learning can recover global structure and interpolate between gaps in the latent space *even without connectedness*. We study this in the Kepler image dataset. In Figure 21, we sampled a training dataset which consisted of four disconnected components together spanning 44% of the phase space, while maintaining partial trajectories with $\alpha = 0.1$. The model is able to interpolate meaningfully on the unseen regions, albeit with some degrading of the representation quality.

It is easier for the model to accommodate the structure of the Kepler dataset then to memorize disconnected components separately. When trying to understand contrastive models, we must incorporate a more subtle understanding of the rich structure of the datasets we analyze in which

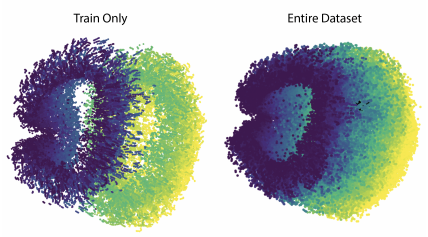

Figure 21: Output embeddings from models trained on 44% of the conserved quantity space and with $\alpha = 0.1$.

we recognize the natural structure of the dataset itself. In particular, it may not be necessary for augmentations within the same class to explicitly overlap, although a similar experiment would be impossible to run directly in CIFAR-10 as it would require reconstruction of the latent space.

I.1 When extrapolation fails

Although there are a wide variety of cases in which the model can learn to interpolate on unseen regions, given too small regions of the phase space, generalizing out-of-distribution will break down. This is visible in extreme distortions of the bowl shape, reversals of trends (e.g. the direction in which H is positive/negative) across different parts of the representation, extremely long "tails" emerging out of the representation, etc., although continuous interpolations on the missing regions will still occur (example Figure 22). At $\alpha = 0.1$, having L and H each take up $2/3$ of the phase space (extreme lower/upper third) tends to be on the boundary between meaningful interpolation and breakdown; introducing any gaps in ϕ_0 tends to cause breakdown. At $\alpha = 1$, it is possible to retrieve the bowl shape with H, L, ϕ_0 each restricted to 30% of the given interval (top/bottom 15%).

The most problematic of all is true extrapolation beyond the input intervals. When we introduce *gaps* in the phase space in which the model must interpolate, it often can do so well. When forced to extrapolate beyond the given phase space, the representation tends to have bizarre behavior (in general, long spikes emerging from the representation). This is visible for, in example, visualizing the representation of the complete dataset (B.5) on a model trained on the normal, incomplete version. This is why in our interpolation experiments we have half the dataset at the upper extreme and the other half at the lower extreme. The model is likely only learning to model necessary functions inside the given intervals; symbolic methods may be superior here in order to avoid this problem.

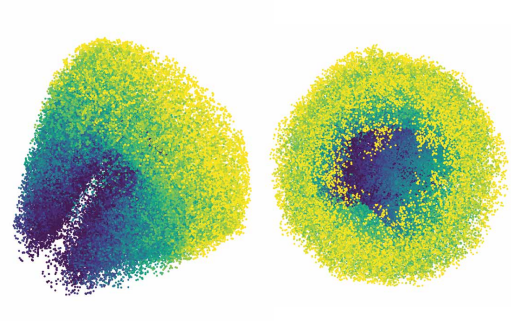

Figure 22: An example of interpolation failure with similar parameters as to Figure 21, colored by L.

J AUGMENTATION RADIUS CONTROLS REPRESENTATION DENSITY

Zimmermann et al. (2021) suggests that contrastively learned representations should be a linear transform of the latent space, in particular, if we let λ be a scale factor for the distribution of positive samples for a single data input, then the scaling factor between the input and output spaces should be proportional to $1/\lambda$ multiplied by a constant not depending on λ, with some conditions.

However, Zimmermann et al. (2021) assumes that the radius of positive examples is uniform over all data samples. When this is not true, we can see increased representation density (and thus decreased representation size) at point where there is more noise, i.e. λ is higher. We can change λ drastically between different parts of the representation to show this phenomenon in action (Figure 23). The possibility of nonuniformity of λ may also have an impact on contrastively learned representations in general. It may apply less to methods like VICreg (Bardes et al., 2022), where variance of representations is explicitly controlled.

Figure 23: Example of selectively increasing λ to shrink parts of the representation. On the right is a normally trained representation with roughly uniform λ; on the left, we added Gaussian noise of scale 0.3 to all three conserved quantities for points on one half of the input domain leading to significant shrinkage.

K BEYOND INTRINSIC DIMENSIONALITY

Since most of our Kepler experiments focused on representations which had the correct dimensionality, we performed some limited analysis on representations with more/less dimensions (Figure 24).

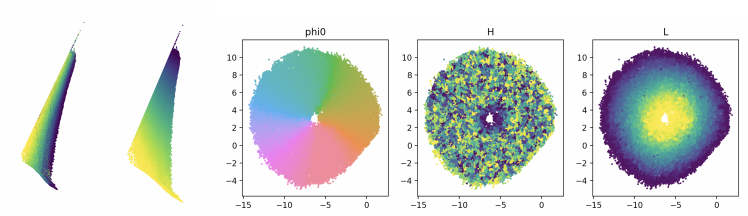

Figure 24: (Left) Too many dimensions (ϕ_0 fixed); colored by H, L in that order. (Right) Too few dimensions.

- With **extra** dimensions (experimental setup: fixing ϕ_0 so that the input is now two-dimensional), the model tends to preserve the correct number of dimensions, causing dimensional collapse (but one that actually corrrectly represents the input space).
- With **not enough** dimensions, the model loses the least important (most dense, see J) conserved quantity, H, but otherwise embeds ϕ_0 and L in a nice-looking 2d projection of the bowl.

L OBSERVED CLUSTERS IN CIFAR-10

Figure 25: Geometrical training progression on projector output of models trained on CIFAR-10 with L2NCE loss; colored by class.

It is unlikely we can observe similarly clear-cut phase transitions when training on CIFAR-10 because of its complexity. However, we can observe repeatable geometric features of the representation which occur during training, which is similar in nature. We trained a version of SimCLR where we forced the projector output to have output dimensionality three (see Hua et al. (2021); Böhm et al. (2022) for similar techniques) and substituted CosNCE for L2NCE. We consistently observed three major stages as training progressed: (1) well-mixed clusters between classes; (2) class separation with noticeable margins; (3) fine-grained cluster emergence within classes (Figure 25). We believe that this appearance is consistent with a series of more local phase changes that together produce these three major phases in training.