Structuring Representation Geometry with Rotationally Equivariant Contrastive Learning

Sharut Gupta[*]
MIT CSAIL
sharut@mit.edu

Joshua Robinson[*]
MIT CSAIL
joshrob@mit.edu

Derek Lim
MIT CSAIL
dereklim@mit.edu

Soledad Villar
Johns Hopkins University
soledad.villar@jhu.edu

Stefanie Jegelka
MIT CSAIL
stefje@csail.mit.edu

Abstract

Self-supervised learning converts raw perceptual data to a compact space using Euclidean distances to measure variations in data. In this paper, we enhance the embedding space by enforcing transformations of input space to correspond to simple (i.e., linear) transformations of embedding space. Specifically, in the contrastive learning setting, we introduce an *equivariance* objective and theoretically prove and empirically demonstrate that its minima forces augmentations on inputs to correspond to *rotations* on the spherical embedding space. Our method, CARE: **C**ontrastive **A**ugmentation-induced **R**otational **E**quivariance, improves performance on downstream tasks by only allowing small rotations.

1 Introduction

Understanding the ideal structure of neural network representation spaces for intelligent behavior to emerge remains limited [Ma et al., 2022]. Learning low-dimensional spaces where simple Euclidean distances effectively measure data similarity is a key factor. Recent advancements have successfully achieved this at web-scale using self-supervision [Chen et al., 2020, Radford et al., 2021]. However, many use cases require richer structural relationships, such as encoding object relations through simple transformations of embeddings, which has driven learning in knowledge graphs [Bordes et al., 2013, Yasunaga et al., 2022]. However, similar capabilities have been notably absent from existing self-supervised learning recipes.

Recent contrastive self-supervised learning approaches have explored ways to close this gap by ensuring input transformations $a \in \mathcal{A}$ correspond to predictable transformations T_a in embedding space i.e., $f(a(x)) \approx T_a f(x)$, a notion called equivariance [Dangovski et al., 2022, Devillers and Lefort, 2023, Garrido et al., 2023, Bhardwaj et al., 2023]. Typically, a learnable feed-forward network is used as T_a, resulting in complex and hard-to-interpret relationships between the embeddings of x and $a(x)$. It also suffers from geometric pathologies, such as inconsistency under compositions: $T_{a_2 \circ a_1} f(x) \neq T_{a_2} T_{a_1} f(x)$.

To address these concerns, we propose CARE, an equivariant contrastive learning framework that learns to approximately translate augmentations in the input space into simple local *linear* transformations in feature space. Considering a hypersphere as our feature space, we consider transformations that are isometries of the sphere: rotations and reflections, i.e., orthogonal transformations. CARE trains f to preserve angles, i.e., $f(a(x))^\top f(a(x')) \approx f(x)^\top f(x')$, a property that must hold if f is

[*]Equal contribution.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

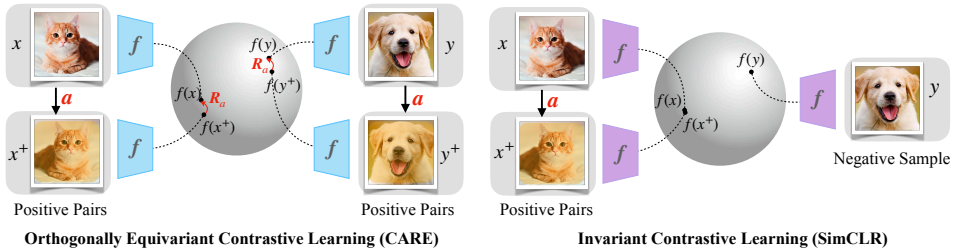

Figure 1: CARE is an equivariant contrastive learning approach that trains augmentations (cropping, blurring, etc.) of input data to correspond to orthogonal transformations of embedding space.

orthogonally equivariant. We show that achieving low error on this seemingly weaker property also implies approximate equivariance and enjoys consistency under compositions. Critically, we can easily integrate CARE into contrastive learning workflows since both operate on pairs of data.

2 Rethinking how augmentations are used in self supervised learning

This work introduces CARE, an equivariant contrastive learning approach respecting two key design principles:

Principle 1. *The map T_a satisfying $f(a(x)) = T_a f(x)$ should be linear, where $f : \mathcal{X} \to \mathbb{S}^{d-1}$ is a feature extracting model mapping to the unit sphere.*

Principle 2. *Equivariance should be learned from* pairs *of data, as in invariant contrastive learning.*

The first principle asks that f converts complex perturbations a of input data into much simpler (i.e., linear) transformations in embedding space. Specifically, we constrain the complexity of T_a by considering isometries of the sphere, $O(d) = \{Q \in \mathbb{R}^{d\times d} : QQ^T = Q^TQ = I\}$, containing all rotations and reflections. Throughout this paper we define $f(a(x)) = T_a f(x)$ for $T_a \in O(d)$ to be *orthogonal equivariance*. This approach draws heavily from ideas in linear representation theory [Curtis and Reiner, 1966, Serre et al., 1977], which studies how to convert abstract group structures into matrix spaces equipped with standard matrix multiplication as the group operation. The second principle stipulates *how* we want to learn orthogonal equivariance. Our method, CARE, explicitly learns T_a by training f so that an augmentation a applied to two different inputs $x, x^+ \in \mathcal{X}$ produces the same change in embedding space. It encodes data augmentations (cropping, blurring, jittering, etc.) as $O(d)$ transformations of embeddings using an equivariance-promoting objective function. CARE can be viewed as an instance of *symmetry regularization* [Shakerinava et al., 2022].

3 CARE: Contrastive Augmentation-induced Rotational Equivariance

This section introduces a simple and practical approach for training a model $f : \mathcal{X} \to \mathbb{S}^{d-1}$ so that f is orthogonally equivariant. To achieve this, we consider the loss $\mathcal{L}_{\text{equi}}(f) = \mathbb{E}_{a\sim\mathcal{A}}\mathbb{E}_{x,x'\sim\mathcal{X}}\left[f(a(x'))^\top f(a(x)) - f(x)^\top f(x')\right]^2$. This is necessarily true if f is orthogonally equivariant or, more generally, $R_a \in O(d)$ exists. But the converse—that $\mathcal{L}_{\text{equi}} = 0$ implies orthogonal equivariance—is non-obvious and is theoretically analyzed in Section 3.1.

A trivial but undesirable solution that minimizes $\mathcal{L}_{\text{equi}}$ is to collapse the embeddings of all points to be the same (see Figure 7). A natural way to do so is to combine the equivariance loss with a non-collapse term such as the uniformity $\mathcal{L}_{\text{unif}}(f) = \log \mathbb{E}_{x,x'\sim\mathcal{X}} \exp\left(f(x)^\top f(x')\right)$ [Wang and Isola, 2020] whose optima f distribute points uniformly over the sphere $\mathcal{L}(f) = \mathcal{L}_{\text{equi}}(f) + \mathcal{L}_{\text{unif}}(f)$. This is directly comparable to the InfoNCE loss, which can similarly be decomposed into two terms $\mathcal{L}_{\text{InfoNCE}}(f) = \mathcal{L}_{\text{inv}}(f) + \mathcal{L}_{\text{unif}}(f)$ where $\mathcal{L}_{\text{inv}}(f) = \mathbb{E}_{a,a'\sim\mathcal{A}}\|f(a(x)) - f(a'(x))\|$ is minimized when f is invariant to $\mathcal{A}$—i.e., $f(a(x)) = f(x)$. Figure 7 shows that training using $\mathcal{L}_{\text{equi}} + \mathcal{L}_{\text{unif}}$ yields non-trivial representations. However, the performance is below that of invariance-based contrastive learning approaches. We hypothesize that this is because data augmentations—which make small perceptual changes to data—should correspond to *small* perturbations of embeddings.

To rule out this possibility, we introduce CARE: **C**ontrastive **A**ugmentation-induced **R**otational **E**quivariance. CARE additionally enforces the orthogonal transformations in embedding space to be *localized* by reintroducing an invariance loss term $\mathcal{L}_{\text{inv}}$ to encourage f to be approximately invariant. Doing so breaks the indifference of $\mathcal{L}_{\text{equi}}$ between large and small rotations, biasing towards small. Specifically, we propose the following objective that combines our equivariant loss with InfoNCE $\mathcal{L}_{\text{CARE}}(f) = \mathcal{L}_{\text{inv}}(f) + \mathcal{L}_{\text{unif}}(f) + \lambda\mathcal{L}_{\text{equi}}(f)$ where λ weights the equivariant loss.

3.1 Theoretical properties of the orthogonally equivariant loss

Proposition 1. *Suppose $\mathcal{L}_{equi}(f) = 0$. Then for almost every $a \in \mathcal{A}$, there is an orthogonal matrix $R_a \in O(d)$ such that $f(a(x)) = R_a f(x)$ for almost all $x \in \mathcal{X}$.*

Figure 1 illustrates this result. This result can be expressed as the existence of a mapping $\rho : \mathcal{A} \to O(d)$ that encodes the space of augmentations within $O(d)$. This raises a natural question: how much of the structure of $\mathcal{A}$ does this encoding preserve?

Corollary 1. *If $\mathcal{L}_{equi}(f) = 0$, then $\rho : \mathcal{A} \to O(d)$ given by $\rho(a) = R_a$ satisfies $\rho(a' \circ a) = \rho(a')\rho(a)$ for almost all a, a'. That is, ρ defines a group action on $\mathbb{S}^{d-1}$ up to a set of measure zero.*

Formally, this result states that if $\mathcal{A}$ is a semi-group, then $\rho : \mathcal{A} \to O(d)$ defines a group homomorphism, or a linear group representation of $\mathcal{A}$ [Curtis and Reiner, 1966]. This property does not hold for non-linear actions [Devillers and Lefort, 2023].

3.2 Extensions to other groups

Notably, the computation of $\mathcal{L}_{\text{equi}}$ solely relies on pairwise data instances $x, x' \in \mathcal{X}$, so it naturally aligns with the contrastive learning paradigm that already works with pairs of data. By changing the inner product, our method applies to other groups that are defined as stabilizers of bilinear forms, such as the Lorentz group, or the symplectic group.

Such extensions to other groups also allow us to use CARE for different embedding space geometries, such as hyperbolic space for self-supervised learners [Ge et al., 2022]. If we constrain our embedding to a hyperboloid model of hyperbolic space, then linear isometries of this space are precisely the Lorentz group. Hence, using our equivariance loss with the Minkowski inner product replacing the Euclidean inner product would allow us to learn hyperbolic representations that transform the embeddings according to the action of the Lorentz group. Further discussions on extensions to other groups and geometries are given in Appendix C.

4 Measuring orthogonal action on embedding space

Wahba's problem. We sample a batch of data $\{x_i\}_{i=1}^n$ and an augmentation a and measure how applying a transforms the embeddings of each x_i consistently. Let F and $F_a \in \mathbb{R}^{d\times n}$ have ith columns $f(x_i)$ and $f(a(x_i))$ respectively, then we compute the error $\mathcal{W}_f = \min_{R\in SO(d)} \|RF - F_a\|_{\text{Fro}}$. Here, $\|\cdot\|_{\text{Fro}}$ represents the Frobenius norm. If $\mathcal{W}_f = 0$, it means that $f(a(x_i)) = R_a f(x_i)$ holds for all i. This problem is widely known as *Wahba's problem*.

Relative rotational equivariance. We define a metric for measuring the equivariance *relative* to the invariance of f, $\gamma_f = \mathbb{E}_{a\sim\mathcal{A}}\mathbb{E}_{x,x'\sim\mathcal{X}}\left\{\frac{(\|f(a(x'))-f(a(x))\|^2-\|f(x')-f(x)\|^2)^2}{(\|f(a(x'))-f(x')\|^2+\|f(a(x))-f(x)\|^2)^2}\right\}$. Details about the metric and the corresponding experimental results are provided in Appendix F.2.1

5 Experiments

5.1 Learning Representations of Protein Point Clouds

We aim to learn protein representations from the Protein Data Bank [Burley et al., 2021] and assess our models by predicting the primary principal component of the protein's point cloud. Given the rotation-equivariant nature of this task, CARE is expected to surpass invariance-centric methods like SimCLR—a claim supported by Figure 2(b). To evaluate rotation equivariance, we sample a

protein and a sequence of rotations along each of the three orthogonal axes. Figure 2(a) visualizes the 2D-projected trajectories of three proteins undergoing three rotation sequences for different training techniques. We find that CARE exhibits a much more regular geometry than models trained with SimCLR, $\mathcal{L}_{\text{unif}}$, or $\mathcal{L}_{\text{inv}}$. Learning the SO(3) manifold is challenging, and previous works assume access to the corresponding group action [Quessard et al., 2020, Park et al., 2021] However, CARE learns it by merely using x and $a(x)$, without relying on the group action a.

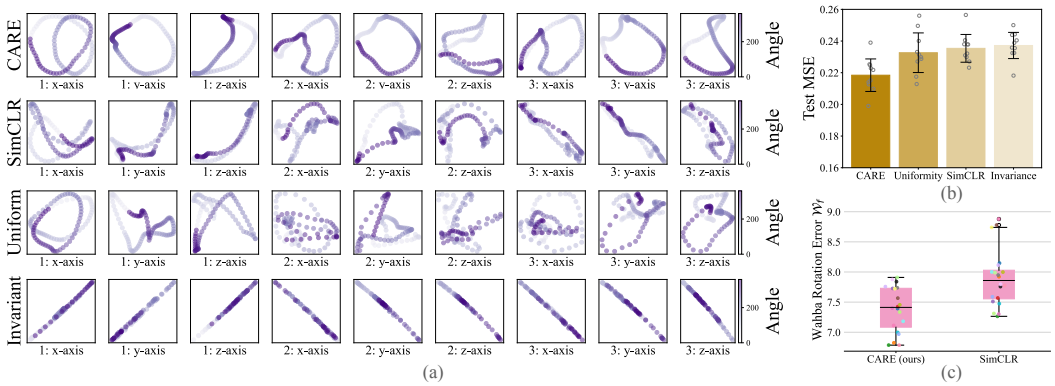

Figure 2: (a) Trajectories through embedding space of three randomly sampled protein point clouds, rotated from 0 to 2π in three orthogonal axes. (b) CARE achieves the lowest error on the task of predicting the first principal component of a protein (c) CARE learns a representation space with better rotational equivariance as it achieves lower error on the Wahba's problem (Section 4).

5.2 Quantitative measures for orthogonal equivariance

Wahba's Problem. We compare ResNet-18 models pretrained with CARE and with SimCLR on CIFAR10. For each model, we compute the optimal value $\mathcal{W}_f$ of Wahba's problem, as introduced in Section 4, over repeated trials. In each trial, we sample a single augmentation $a \sim \mathcal{A}$ at random and compute $\mathcal{W}_f$ for $f = f_{\text{CARE}}$ and $f = f_{\text{SimCLR}}$ over the test data. We repeat this process 20 times and plot the results in Figure 2(c), where the colors of dots indicate the sampled augmentation. Results show that CARE has a lower average error and worst-case error. Furthermore, comparing point-wise for a single augmentation, CARE achieves lower error in nearly all cases.

Results for relative rotational equivariance metric are reported in Appendix F.2.1

Ablation of loss terms. The CARE loss $\mathcal{L}_{\text{CARE}}$ is a weighted sum of the InfoNCE loss $\mathcal{L}_{\text{InfoNCE}}$ and the orthogonal equivariance loss $\mathcal{L}_{\text{equi}}$. Figure 8 evaluates the performance of ResNet-50 models trained on CIFAR10 using $\mathcal{L}_{\text{InfoNCE}} + \lambda\mathcal{L}_{\text{equi}}$ for varying λ, finding optimal λ in the range $0.01 \leq \lambda \leq 0.1$. Additional ablations of combinations of the three losses ($\mathcal{L}_{\text{inv}}$, $\mathcal{L}_{\text{unif}}$, $\mathcal{L}_{\text{equi}}$) are reported in Figure 7.

Results for qualitative measures are presented in Appendix F.2.1 and E, respectively.

Table 1: Top-1 linear probe accuracy (%) on CIFAR10, CIFAR100, STL10 and ImageNet100 datasets. We report the mean performance from 3 different random initializations for the linear classifer. * denote numbers from Devillers and Lefort [2023], and ** from Zhuo et al. [2023]

Method	CIFAR10	CIFAR100	STL10	ImageNet100
Invariant prediction approaches				
SimCLR	90.98	66.77	84.19	72.79
MoCo-v2	91.95	69.88	-	73.50
BYOL	90.44*	67.41**	-	-
Equivariant prediction approaches				
EquiMod$_{\text{SimCLR}}$	91.28	67.59	83.67	-
EquiMod$_{\text{BYOL}}$	91.57*	-	-	-
CARE$_{\text{SimCLR}}$	**91.92** (↑ 0.94)	**68.05** (↑ 1.28)	**84.64** (↑ 0.45)	**76.69** (↑ 3.90)
CARE$_{\text{MoCo-v2}}$	**92.19** (↑ 0.24)	**70.56** (↑ 0.68)	**88.97**	**74.30** (↑ 0.80)

5.3 Linear probe for image classification

We examine the quality of features learned by CARE for solving image classification tasks on four benchmarks: CIFAR10, CIFAR100, STL10, and ImageNet100 (see Appendix E for details). Table 1 shows consistent improvements in performance using CARE, showing the benefits of our structured embedding approach for image recognition tasks.

Detailed discussion about the limitations and broader impact of our work is provided in Appendix G.

6 Acknowledgements

This research was supported by NSF award CCF-2112665. Derek Lim is supported by National Science Foundation Graduate Research Fellowship. Soledad Villar is partially funded by the NSF–Simons Research Collaboration on the Mathematical and Scientific Foundations of Deep Learning (MoDL) (NSF DMS 2031985), NSF CISE 2212457, ONR N00014-22-1-2126 and an Amazon AI2AI Faculty Research Award. We acknowledge MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing HPC resources that have contributed to this work.

References

Sangnie Bhardwaj, Willie McClinton, Tongzhou Wang, Guillaume Lajoie, Chen Sun, Phillip Isola, and Dilip Krishnan. Steerable equivariant representation learning. *preprint arXiv:2302.11349*, 2023.

Ben Blum-Smith and Soledad Villar. Equivariant maps from invariant functions. *preprint arXiv:2209.14991*, 2022.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 26, 2013.

Stephen K Burley, Charmi Bhikadiya, Chunxiao Bi, Sebastian Bittrich, Li Chen, Gregg V Crichlow, Cole H Christie, Kenneth Dalenberg, Luigi Di Costanzo, Jose M Duarte, et al. Rcsb protein data bank: powerful new tools for exploring 3d structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. *Nucleic acids research*, 49(D1):D437–D451, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pages 1597–1607. PMLR, 2020.

Charles W Curtis and Irving Reiner. *Representation theory of finite groups and associative algebras*, volume 356. American Mathematical Soc., 1966.

Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljačić. Equivariant contrastive learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Alexandre Devillers and Mathieu Lefort. Equimod: An equivariance module to improve self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2023.

Quentin Garrido, Laurent Najman, and Yann Lecun. Self-supervised learning of split invariant equivariant representations. *preprint arXiv:2302.10283*, 2023.

Songwei Ge, Shlok Mishra, Simon Kornblith, Chun-Liang Li, and David Jacobs. Hyperbolic contrastive learning for visual representations beyond objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Yi Ma, Doris Tsao, and Heung-Yeung Shum. On the principles of parsimony and self-consistency for the emergence of intelligence. *Frontiers of Information Technology & Electronic Engineering*, 23 (9):1298–1323, 2022.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

Jung Yeon Park, Ondrej Biza, Linfeng Zhao, Jan-Willem van de Meent, and Robin Walters. Learning symmetric representations for equivariant world models. 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Robin Quessard, Thomas D Barrett, and William R Clements. Learning group structure and disentangled representations of dynamical environments. *arXiv preprint arXiv:2002.06991*, 2020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021.

Barbara J Schmid. Finite groups and invariant theory. In *Topics in Invariant Theory: Séminaire d'Algèbre P. Dubreil et M.-P. Malliavin 1989–1990 (40ème Année)*, pages 35–66. Springer, 2006.

Jean-Pierre Serre et al. *Linear representations of finite groups*, volume 42. Springer, 1977.

Mehran Shakerinava, Arnab Kumar Mondal, and Siamak Ravanbakhsh. Structuring representations using group invariants. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Soledad Villar, David W Hogg, Kate Storey-Fisher, Weichi Yao, and Ben Blum-Smith. Scalars are universal: Equivariant machine learning, structured like classical physics. pages 28848–28863, 2021.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning (ICML)*, pages 9929–9939. PMLR, 2020.

Hermann Weyl. *The classical groups: their invariants and representations*. Princeton university press, 1946.

Keyulu Xu, Mozhi Zhang, Jingling Li, Simon S Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. How neural networks extrapolate: From feedforward to graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2021.

Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D Manning, Percy S Liang, and Jure Leskovec. Deep bidirectional language-knowledge graph pretraining. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 37309–37323, 2022.

Zhijian Zhuo, Yifei Wang, Jinwen Ma, and Yisen Wang. Towards a unified theoretical understanding of non-contrastive learning via rank differential mechanism. In *International Conference on Learning Representations (ICLR)*, 2023.

A Proofs of theoretical results

The aim of this section is to detail the proofs of the theoretical results presented in the main manuscript. The key theoretical tools driving our analysis are prepared separately in Section B.

Throughout our analysis, we assume that all spaces (e.g., $\mathcal{A}$ and $\mathcal{X}$) are subspaces of Euclidean space and therefore admit a Lebesgue measure. We also assume that all distributions (e.g., $a \sim \mathcal{A}$ and $x \sim \mathcal{X}$) admit a density with respect to the Lebesgue measure. With these conditions in mind, we recall the loss function that is the main object of study:

$$\mathcal{L}_{\text{equi}}(f) = \mathbb{E}_{a\sim\mathcal{A}}\mathbb{E}_{x,x'\sim\mathcal{X}}\left[f(a(x'))^\top f(a(x)) - f(x)^\top f(x')\right]^2 \tag{1}$$

Next, we re-state and prove Proposition 1, our first key result.

Proposition 1. *Suppose $\mathcal{L}_{equi}(f) = 0$. Then for almost every $a \in \mathcal{A}$, there is an orthogonal matrix $R_a \in O(d)$ such that $f(a(x)) = R_a f(x)$ for almost all $x \in \mathcal{X}$.*

Proof. Suppose that $\mathcal{L}_{\text{equi}}(f) = 0$. This means that $f(a(x'))^\top f(a(x)) = f(x)^\top f(x')$ for almost all $a \in G$, and $x, x' \in \mathcal{X}$. Setting $g_a(x) = f(a(x))$, we have that $g_a(x')^\top g_a(x) = f(x)^\top f(x')$. The continuous version of the First Fundamental Theorem of invariant theory for the orthogonal group (see Proposition 4) implies that there is an $R_a \in O(d)$ such that $f(a(x)) = g_a(x) = R_a f(x)$. □

As discussed in greater detail in the main manuscript, these results show that minimizing $\mathcal{L}_{\text{equi}}$ produces a model where an augmentation a corresponds to a single orthogonal transformation of embeddings R_a, independent of the input. This result is continuous in flavor as it studies the loss over the full data distribution $p(x)$. There exists a corresponding result for the finite sample loss

$$\mathcal{L}_{\text{equi},n}(f) = \mathbb{E}_{a\sim\mathcal{A}} \sum_{i,j=1}^{n} \left[f(a(x_j))^\top f(a(x_i)) - f(x_i)^\top f(x_j)\right]^2.$$

Proposition 2. *Suppose $\mathcal{L}_{equi,n}(f) = 0$. Then for almost every $a \in \mathcal{A}$, there is an orthogonal matrix $R_a \in O(d)$ such that $f(a(x_i)) = R_a f(x_i)$ for all $i = 1, \dots, n$.*

As for the population counterpart, the proof of this result directly follows from the application of the First Fundamental Theorem of invariant theory for the orthogonal group.

Proof of Proposition 2. Suppose that $\mathcal{L}_{\text{equi}}(f) = 0$. This means that for almost every $a \in G$, and every $i, j = 1, \dots, n$ we have $f(a(x_j))^\top f(a(x_i)) = f(x_i)^\top f(x_j)$. In other words $AA^T = BB^T$ where $A, B \in \mathbb{R}^{n\times d}$ are matrices whose ith rows are $A_i = f(a(x_i))^\top$ and $B_i = f(x_i)^\top$ respectively. This implies, by the First Fundamental Theorem of invariant theory for the orthogonal group (see Corollary 2), that there is an $R_a \in O(d)$ such that $A = BR_a$. Considering only the ith rows of A and B leads us to conclude that $f(a(x_i)) = R_a f(x_i)$. □

A corollary of Proposition 1 is that compositions of augmentations correspond to compositions of rotations.

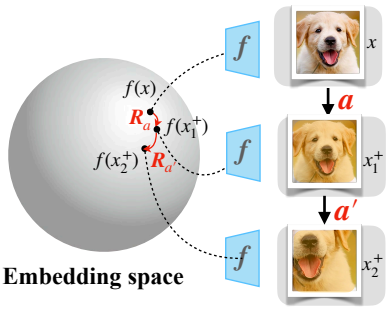

Figure 3: When $\mathcal{L}_{\text{equi}} = 0$, compositions of augmentations correspond to compositions of rotations.

Corollary 1. *If $\mathcal{L}_{equi}(f) = 0$, then $\rho : \mathcal{A} \to O(d)$ given by $\rho(a) = R_a$ satisfies $\rho(a' \circ a) = \rho(a')\rho(a)$ for almost all a, a'. That is, ρ defines a group action on $\mathbb{S}^{d-1}$ up to a set of measure zero.*

Proof. Applying Proposition 1 on $a' \circ a$ as the sampled augmentation, we have that $f(a' \circ a(x_i)) = R_{a' \circ a} f(x_i) = \rho(a' \circ a) f(x_i)$. However, taking $\bar{x} = a(x_i)$ and applying Proposition 1 twice we also know that $f(a' \circ a(x_i)) = f(a'(\bar{x})) = R_a f(\bar{x}) = R_{a'} f(a(x_i)) = R_{a'} R_a f(x) = \rho(a')\rho(a) f(x_i)$. That is, $\rho(a' \circ a) f(x_i) = f(a' \circ a(x_i)) = \rho(a')\rho(a) f(x_i)$. Since this holds for all i, we have that $\rho(a' \circ a) = \rho(a')\rho(a)$. □

This corollary requires us to assume that $\mathcal{A}$ is a semi-group. That is, $\mathcal{A}$ is closed under compositions, but group elements do not necessarily have inverses and it does not need to include an identity element.

B Background on invariance theory for the orthogonal group

This section recalls some classical theory on orthogonal groups and an extension that we use for proving results over continuous data distributions.

A function $f : (\mathbb{R}^d)^n \to \mathbb{R}$ is said to be $O(d)$-invariant if $f(Rv_1, \ldots, Rv_n) = f(v_1, \ldots, v_n)$ for all $R \in O(d)$. Throughout this section, we are especially interested in determining easily computed statistics that *characterize* an $O(d)$ invariant function f. In other words, we would like to write f as a function of these statistics. The following theorem was first proved by Hermann Weyl using Capelli's identity [Weyl, 1946] and shows that the inner products $v_i^\top v_j$ suffice.

Theorem 3 (First fundamental theorem of invariant theory for the orthogonal group)**.** *Suppose that $f : (\mathbb{R}^d)^n \to \mathbb{R}$ is $O(d)$-invariant. Then there exists a function $g : \mathbb{R}^{n \times n} \to \mathbb{R}$ for which*

$$f(v_1, \ldots, v_n) = g\big([v_i^\top v_j]_{i,j=1}^n\big).$$

In other words, to compute f at a given input, it is not necessary to know all of $v_1, \ldots, v_n$. Computing the value of f at a point can be done using only the inner products $v_i^\top v_j$, which are invariant to $O(d)$. Letting V be the $n \times d$ matrix whose ith row is $v_i^\top$, we may also write $f(v_1, \ldots, n_n) = g(VV^\top)$. The map $V \mapsto VV^\top$ is known as the orthogonal projection of V.

A corollary of this result has recently been used to develop $O(d)$ equivariant architectures in machine learning [Villar et al., 2021].

Corollary 2. *Suppose that A, B are $n \times d$ matrices and $AA^\top = BB^\top$. Then $A = BR$ for some $R \in O(d)$.*

Villar et al. [2021] use this characterization of orthogonally equivariant functions to *parameterize* function classes of neural networks that have the same equivariance. This result is also useful in our context; However, we put it to use for a very different purpose: studying $\mathcal{L}_{\text{equi}}$.

Intuitively this result says the following: given two point clouds A, B of unit length vectors with some fixed correspondence (bijection) between each point in A and a point in B, if the *angles* between the ith and jth points in cloud A always equal the angle between the ith and jth point in cloud B, then A and B are the same up to an orthogonal transformation.

This is the main tool we use to prove the finite sample version of the main result for our equivariant loss (Proposition 2). However, to analyze the population sample loss $\mathcal{L}_{\text{equi}}$ (Proposition 1), we require an extended version of this result to the continuous limit as $n \to \infty$. To this end, we develop a simple but novel extension to Theorem 3 to the case of continuous data distributions. This result may be useful in other contexts independent of our setting.

Proposition 4. *Let $\mathcal{X}$ be any set and $f, h : \mathcal{X} \to \mathbb{R}^d$ be functions on $\mathcal{X}$. If $f(x)^\top f(y) = h(x)^\top h(y)$ for all $x, y \in \mathcal{X}$, then there exists $R \in O(d)$ such that $Rf(x) = h(x)$ for all $x \in \mathcal{X}$.*

The proof of this result directly builds on the finite sample version. The key idea of the proof is that since the embedding space $\mathbb{R}^d$ is finite-dimensional we may select a set of points $\{f(x_i)\}_i$ whose span has maximal rank in the linear space spanned by the outputs of f. This means that any arbitrary point $f(x)$ can be written as a linear combination of the $f(x_i)$. This observation allows us to apply the finite sample result on each $f(x_i)$ term in the sum to conclude that $f(x)$ is also a rotation of a sum of $h(x_i)$ terms. Next, we give the formal proof.

Proof of Proposition 4. Choose $x_1, \ldots, x_n \in \mathcal{X}$ such that $F = [f(x_1) \mid \ldots \mid f(x_n)]^\top \in \mathbb{R}^{n \times d}$ and $h = [h(x_1) \mid \ldots \mid h(x_n)]^\top \in \mathbb{R}^{n \times d}$ have maximal rank. Note we use "$\mid$" to denote the column-wise concatenation of vectors. Note that such x_i can always be chosen. Since we have $FF^\top = HH^\top$, we know by Corollary 2 that $F = HR$ for some $R \in O(d)$.

Now consider an arbitrary $x \in \mathcal{X}$ and define $\tilde{F} = [F \mid f(x)]^\top$ and $\tilde{H} = [H \mid h(x)]^\top$, both of which belong to $\mathbb{R}^{(n+1)\times d}$. Note that again we have $\tilde{F}\tilde{F}^\top = \tilde{H}\tilde{H}^\top$ so also know that $\tilde{F} = \tilde{H}\tilde{R}$ for some $\tilde{R} \in O(d)$. Since x_i were chosen so that F and H are of maximal rank, we know that $h(x) = \sum_{i=1}^n c_i h(x_i)$ for some coefficients $c_i \in \mathbb{R}$, since if this were not the case then we would have $\text{rank}(\tilde{H}) = \text{rank}(H) + 1$.

From this, we know that

$$\begin{aligned} R^\top h(x) &= \sum_{i=1}^n c_i R^\top h(x_i) \\ &= \sum_{i=1}^n c_i f(x_i) \\ &= \sum_{i=1}^n c_i \tilde{R}^\top h(x_i) \\ &= \tilde{R}^\top \sum_{i=1}^n c_i h(x_i) \\ &= \tilde{R}^\top h(x) \\ &= f(x). \end{aligned}$$

So we have that $Rf(x) = RR^\top h(x) = h(x)$ for all $x \in \mathcal{X}$. □

C Extensions to other groups: further discussion

In Section 3.2, we explore the possibility of formulating an equivariant loss $\mathcal{L}_{\text{equi}}$ for pairs of points that fully captures equivariance by requiring the group to be the stabilizer of a bilinear form. In this context, the invariants are generated by polynomials of degree two in two variables, and the equivariant functions can be obtained by computing gradients of these invariants [Blum-Smith and Villar, 2022]. Section 3.2 notes that this holds true not only for the orthogonal group, which is the primary focus of our research but also for the Lorentz group and the symplectic group, suggesting natural extensions of our approach.

It is worth noting that the group of rotations $SO(d)$ does not fall into this framework. It can be defined as the set of transformations that preserve both inner products (a 2-form) and determinants (a d-form). Consequently, some of its generators have degree 2 while others have degree d (see [Weyl, 1946], Section II.A.9).

Weyl's theorem states that if a group acts on n copies of a vector space (in our case, $(\mathbb{R}^d)^n$ for consistency with the rest of the paper), its action can be characterized by examining how it acts on k copies (i.e., $(\mathbb{R}^d)^k$) when the maximum degree of its irreducible components is k (refer to Section 6 of [Schmid, 2006] for a precise statement of the theorem). Since our interest lies in understanding equivariance in terms of pairs of objects, we desire invariants that act on pairs of points. One way to guarantee this is to restrict ourselves to groups that act through representations where the irreducible components have degrees of at most two (though this is not necessary in all cases, such as the orthogonal group $O(d)$ that we consider in the main paper). An example of such groups is the product of finite subgroups of the unitary group $U(2)$, which holds relevance in particle physics. According to Weyl's theorem, the corresponding invariants can be expressed as *polarizations* of degree-2 polynomials on two variables. Polarizations represent an algebraic construction that enables the expression of homogeneous polynomials in multiple variables by introducing additional variables to polynomials with fewer variables. In our case, the base polynomials consist of degree-2

polynomials in two variables, while the polarizations incorporate additional variables. Notably, an interesting open problem lies in leveraging this formulation for contrastive learning.

D Implementation details

Algorithm 1 presents pytorch-based pseudocode for implementing CARE. This implementation introduces the idea of using a smaller batch size for the equivariance loss compared to the InfoNCE loss. Specifically, by definition, the equivariance loss is defined as a double expectation, one over data pairs and the other over augmentations. Empirical observations reveal that sampling one augmentation per batch leads to unstable yet superior performance when compared to standard invariant-based baselines such as SimCLR. Since these invariant-based contrastive benchmarks generally perform well with large batch sizes, we adopt the approach of splitting a batch into multiple chunks to efficiently sample multiple augmentations per batch for the equivariance loss. Each chunk of the batch is associated with a new pair of augmentations, ensuring a large batch size for the InfoNCE loss and a smaller batch size for the equivariance loss.

Algorithm 1 PyTorch based pseudocode for CARE

Notations: f represents the backbone encoder network, λ is the weight on CARE loss, `apply_same_aug` function applies the same augmentation to all samples in the input batch
for minibatch x in `dataloader` **do**
  draw *two batches* of augmentation functions $a_1, a_2 \in \mathcal{A}$
  /* Functions a_1, a_2 apply different augmentation to each sample in batch x */
  $z_1^{\text{inv}}, z_2^{\text{inv}} = f(a_1(x)), f(a_2(x))$
  divide x into `n_split` chunks to form x_{chunks}
  /* Module for calculating orthogonal equivariance loss */
  for c_i in x_{chunks} in parallel **do**
    draw *two* augmentation functions $\tilde{a}_1, \tilde{a}_2 \in \mathcal{A}$
    /* Functions $\tilde{a}_1$, $\tilde{a}_2$ apply same augmentation to each sample in batch c_i */
    $\tilde{z}_{i1}, \tilde{z}_{i2} = f(\texttt{apply\_same\_aug}(c_i, \tilde{a}_1)), f(\texttt{apply\_same\_aug}(c_i, \tilde{a}_2))$
  /* Concatenate embedding vectors corresponding to all chunks */
  merge $\tilde{z}_{i1}, \tilde{z}_{i2}$ into $z_1^{\text{equiv}}, z_2^{\text{equiv}}$ respectively
  /* Loss computation */
  $\mathcal{L}_{\text{InfoNCE}}(f) = \texttt{infonce\_loss}(z_1^{\text{inv}}, z_2^{\text{inv}})$
  $\mathcal{L}_{\text{equiv}}(f) = \texttt{orthogonal\_equivariance\_loss}(z_1^{\text{equiv}}, z_2^{\text{equiv}}, \texttt{n\_split})$
  $\mathcal{L}_{\text{CARE}}(f) = \mathcal{L}_{\text{InfoNCE}}(f) + \lambda \cdot \mathcal{L}_{\text{equiv}}(f)$
  /* Optimization step */
  $\mathcal{L}_{\text{CARE}}(f)$.backward()
  optimizer.step()

E Supplementary experimental details and assets disclosure

E.1 Assets

We do not introduce new data in the course of this work. Instead, we use publicly available widely used image datasets for the purposes of benchmarking and comparison.

E.2 Hardware and setup

All experiments were performed on an HPC computing cluster using 4 NVIDIA Tesla V100 GPUs with 32GB accelerator RAM for a single training run. The CPUs used were Intel Xeon Gold 6248 processors with 40 cores and 384GB RAM. All experiments use the PyTorch deep learning framework [Paszke et al., 2019].

E.3 Experimental protocols

We first outline the training protocol adopted for training our proposed approach on a variety of datasets, namely CIFAR10, CIFAR100, STL10, and ImageNet100.

CIFAR10, CIFAR100 and STL10 All encoders have ResNet-50 backbones and are trained for 400 epochs with temperature $\tau = 0.5$ for SimCLR and $\tau = 0.1$ for MoCo-v2 [2]. The encoded features have a dimension of 2048 and are further processed by a two-layer MLP projection head, producing an output dimension of 128. A batch size of 256 was used for all datasets. For CIFAR10 and CIFAR100, we employed the Adam optimizer with a learning rate of $1e^{-3}$ and weight decay of $1e^{-6}$. For STL10, we employed the SGD optimizer with a learning rate of 0.06, utilizing cosine annealing and a weight decay of $5e^{-4}$, with 10 warmup steps. We use the same set of augmentations as in SimCLR [Chen et al., 2020]. To train the encoder using $\mathcal{L}_{\text{CARE-SimCLR}}$, we use the same hyper-parameters for InfoNCE loss. Additionally, we use 4, 8 and 16 batch splits for CIFAR100, STL10 and CIFAR10, respectively. This allows us to sample multiple augmentations per batch, effectively reducing the batch size of equivariance loss whilst retaining the same for InfoNCE loss. Furthermore, for the equivariant term, we find it optimal to use a weight of $\lambda = 0.01, 0.001$, and 0.01 for CIFAR10, CIFAR100, and STL10, respectively.

ImageNet100 We use ResNet-50 as the encoder architecture and pretrain the model for 200 epochs. A base learning rate of 0.8 is used in combination with cosine annealing scheduling and a batch size of 512. For MoCo-v2, we use 0.99 as the momentum and $\tau = 0.2$ as the temperature. All remaining hyperparameters were maintained at their respective official defaults as in the official MoCo-v2 code. While training with $\mathcal{L}_{\text{CARE-SimCLR}}$ and $\mathcal{L}_{\text{CARE-MoCo}}$, we find it optimal to use splits of 4 and 8 and weight of $\lambda = 0.005$ and 0.01 respectively on the equivariant term.

Linear evaluation We train a linear classifier on frozen features for 100 epochs with a batch size of 512 for CIFAR10, CIFAR100, and STL10 datasets. To optimize the classifier, we employ the Adam optimizer with a learning rate of $1e^{-3}$ and a weight decay of $1e^{-6}$. In the case of ImageNet100, we train the linear classifier for 60 epochs using a batch size of 128. We initialize the learning rate to 30.0 and apply a step scheduler with an annealing rate of 0.1 at epochs 30, 40, and 50. The remaining hyper-parameters are retained from the official code.

F Additional experiments

F.1 Qualitative assessment of equivariance

A key property promised by equivariant contrastive models is sensitivity to specific augmentations. To qualitatively evaluate the sensitivity, or equivariance, of our models, we consider an image retrieval task on the Flowers-102 dataset [Nilsback and Zisserman, 2008], as considered by [Bhardwaj et al., 2023]. Specifically, when presented with an input image x, we extract the top 5 nearest neighbors based on the Euclidean distance of $f(x)$ and $f(a(x))$, where $a \in \mathcal{A}$. We report the results of using color jitter as a transformation of the input, comparing the invariant (SimCLR) and our equivariant (CARE) models in Figure 4. We see that retrieved results for the CARE model exhibit greater variability in response to a change in query color compared to the SimCLR model. Notably, the color of the retrieved results for all queries in the SimCLR model remains largely invariant, thereby confirming its robustness to color changes.

F.2 Quantitative assessment of equivariance

F.2.1 Relative rotational equivariance.

Optimizing for the CARE objective may potentially result in learning invariance rather than equivariance. Specifically, for input image x, $f(a(x)) = f(x)$ for $a \in \mathcal{A}$ is a trivial optimal solution of $\arg\min_f \mathcal{L}_{\text{equi}}(f)$. To check that our model is learning non-trivial equivariance, we consider a metric similar to one proposed by [Bhardwaj et al., 2023] for measuring the equivariance *relative* to the invariance of f:

$$\gamma_f = \mathbb{E}_{a\sim\mathcal{A}}\mathbb{E}_{x,x'\sim\mathcal{X}}\left\{\frac{(\|f(a(x')) - f(a(x))\|^2 - \|f(x') - f(x)\|^2)^2}{(\|f(a(x')) - f(x')\|^2 + \|f(a(x)) - f(x)\|^2)^2}\right\}. \tag{2}$$

Here, the denominator measures the invariance of the representation, with smaller values corresponding to greater invariance to the augmentations. The numerator, on the other hand, measures

[2] https://github.com/facebookresearch/moco

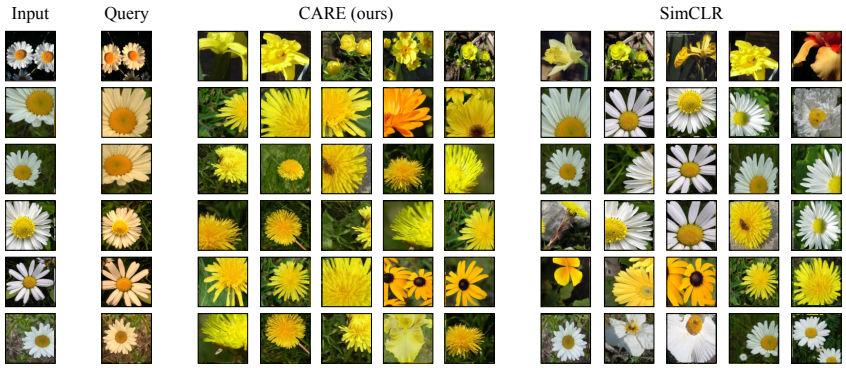

Figure 4: CARE exhibits sensitivity to features that invariance-based contrastive methods (e.g., SimCLR) do not. For each input we apply color jitter to produce the query image. We then retrieve the 5 nearest neighbors in the embedding space of CARE and SimCLR.

equivariance and can be simplified to $\left[f(a(x'))^\top f(a(x)) - f(x)^\top f(x')\right]^2$ (i.e., $\mathcal{L}_{\text{equi}}(f)$) up to a constant, because f maps to the unit sphere. The ratio γ_f of these two terms measures the non-trivial equivariance, with a lower value implying greater non-trivial orthogonal equivariance.

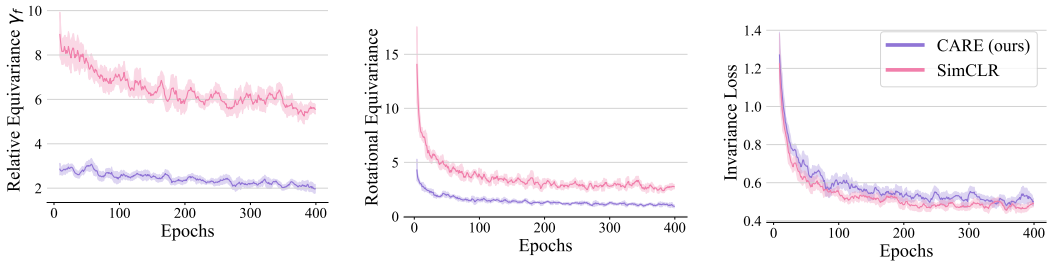

Figure 5: **Relative rotational equivariance** (lower is more equivariant). Both CARE and invariance-based contrastive methods (e.g., SimCLR) produce *approximately* invariant embeddings. However, they differ in their residual sensitivity to augmentations. CARE learns a considerably more rotationally structured embedding space. We note that this is in part because CARE is less invariant to augmentations (higher invariance loss).

We measure the relative rotational equivariance for both CARE and SimCLR over the course of pretraining by following the approach outlined in Section 4. Specifically, we compare ResNet-18 models trained using CARE and SimCLR on CIFAR10. From Figure 5, we observe that both the models produce embeddings with comparable non-zero invariance loss $\mathcal{L}_{\text{inv}}$, indicating approximate invariance. However, they differ in their sensitivity to augmentations, with CARE attaining a much lower relative equivariance error. Importantly, this shows that CARE is *not* achieving lower equivariance error $\mathcal{L}_{\text{equi}}$ by collapsing to invariance, a trivial form of equivariance.

F.2.2 Analyzing structure on a 2D manifold.

To further study $\mathcal{L}_{\text{equi}}$, we train an encoder f that projects the input onto $\mathbb{S}^1$, the unit circle in the 2D plane. In this case, orthogonal transformations are characterized by *angles*. We sample an augmentation $a \sim \mathcal{A}$ and measure the cosine of the angle between pairs $f(x)$ and $f(a(x))$ for all x in the test set. This process is repeated for 20 distinct sampled augmentations, and the density of all recorded cosine angles is recorded in Figure 6. Both CARE and SimCLR exhibit high density close to 1, demonstrating approximate invariance. However, unlike CARE, SimCLR exhibits non-zero density in the region -0.5 to -1.0, indicating that the application of augmentations significantly displaces the embeddings. Additionally, CARE consistently exhibits lower variance σ^2 of the cosine angles between $f(x)$ and $f(a(x))$ for a fixed augmentation, as expected given that it is supposed to transform all embeddings in the same way.

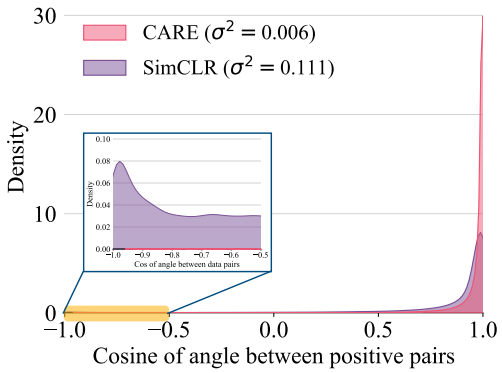

Figure 6: Histogram of the cosine of angles between data pairs for CARE and SimCLR. CARE exhibits a significantly lower variance of cosine similarity values compared to SimCLR.

F.3 Ablating loss terms

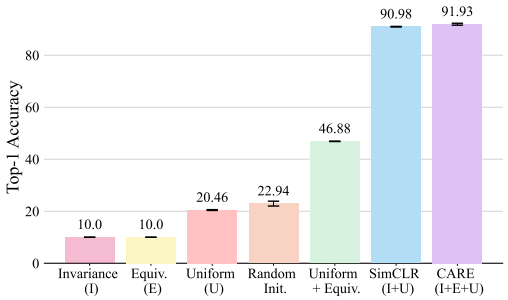

Figure 7: Ablating different loss terms. Combining $\mathcal{L}_{\text{equi}}$ with a uniformity promoting non-collapse term suffices to learn non-trivial features. However, optimal performance is achieved when encouraging *smaller* rotations, as in CARE. ResNet-50 models pretrained on CIFAR10 and evaluated with linear probes.

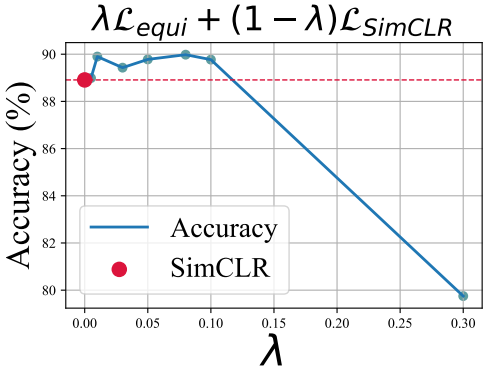

Figure 8: Linear readout error as the loss weightings vary.

Histogram for loss ablation. To accompany Figure 7, this section plots the cosine similarity between positive pairs. We provide two plots for each experiment: the first plots the *histogram* of similarities of positive pairs drawn from the test set; the second plots the *average* positive cosine similarity throughout training. The results are reported in Figures 9, 10, 11, 12, 13, 14.

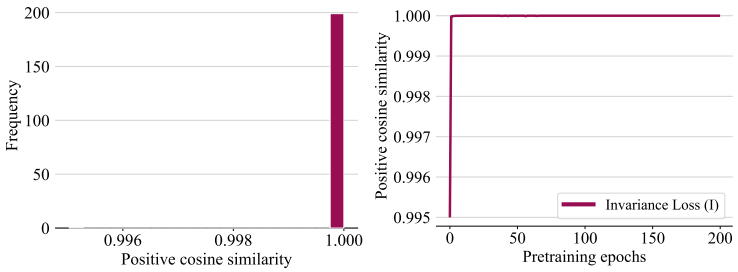

Figure 9: (left) Histogram of positive cosine similarity values at the end of pre-training using the invariance loss; (right) Evolution of positive cosine similarity values over pre-training epochs using the invariance loss

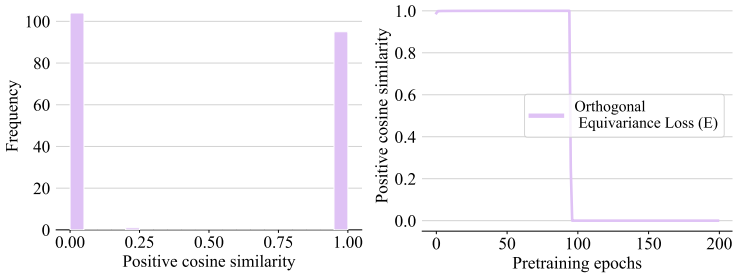

Figure 10: (left) Histogram of positive cosine similarity values at the end of pre-training using the orthogonal equivariance loss; (right) Evolution of positive cosine similarity values over pre-training epochs using the orthogonal equivariance loss

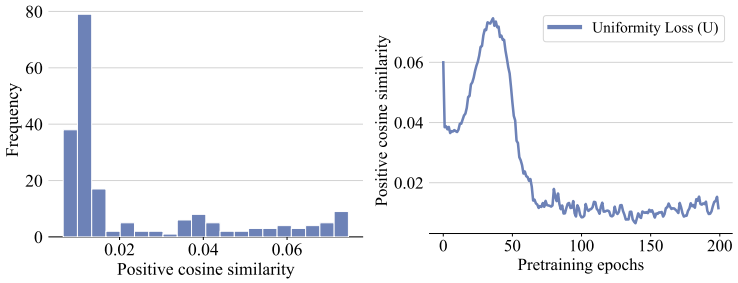

Figure 11: (left) Histogram of positive cosine similarity values at the end of pre-training using the uniformity loss; (right) Evolution of positive cosine similarity values over pre-training epochs using the uniformity loss

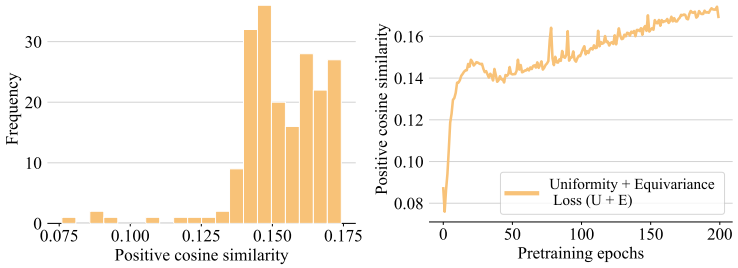

Figure 12: (left) Histogram of positive cosine similarity values at the end of pre-training using the Uniformity + Equivariance loss; (right) Evolution of positive cosine similarity values over pre-training epochs using the Uniformity + Equivariance loss

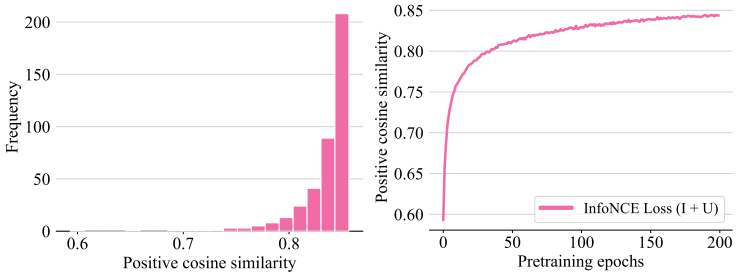

Figure 13: (left) Histogram of positive cosine similarity values at the end of pre-training using the InfoNCE (invariance + uniformity) loss; (right) Evolution of positive cosine similarity values over pre-training epochs using the InfoNCE loss

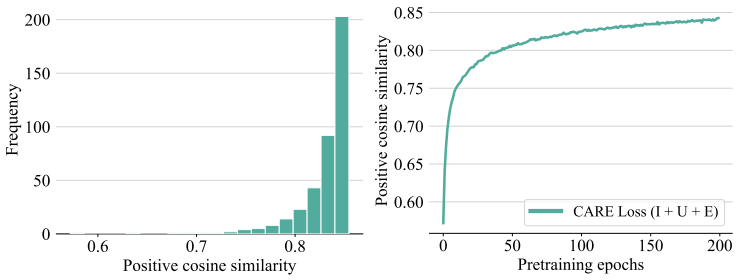

Figure 14: (left) Histogram of positive cosine similarity values at the end of pre-training using the CARE (InfoNCE + orthogonal equivariance) loss; (right) Evolution of positive cosine similarity values over pre-training epochs using the CARE loss

F.4 Additional Protein Trajectories

Figures 15, 16, 17 and 18 illustrate additional trajectories observed through the embedding space of a DeepSet trained with the CARE, SimCLR, $\mathcal{L}_{\text{unif}}$ and $\mathcal{L}_{\text{inv}}$ loss respectively.

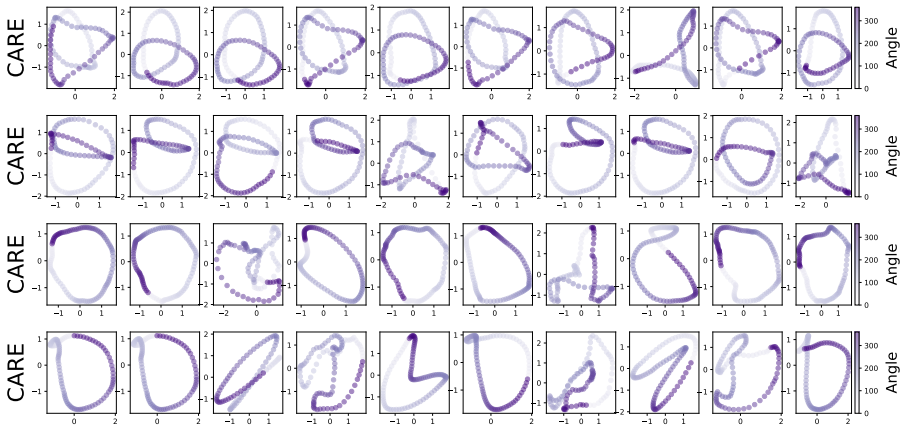

Figure 15: Additional trajectories through the embedding space of a DeepSet trained with CARE.

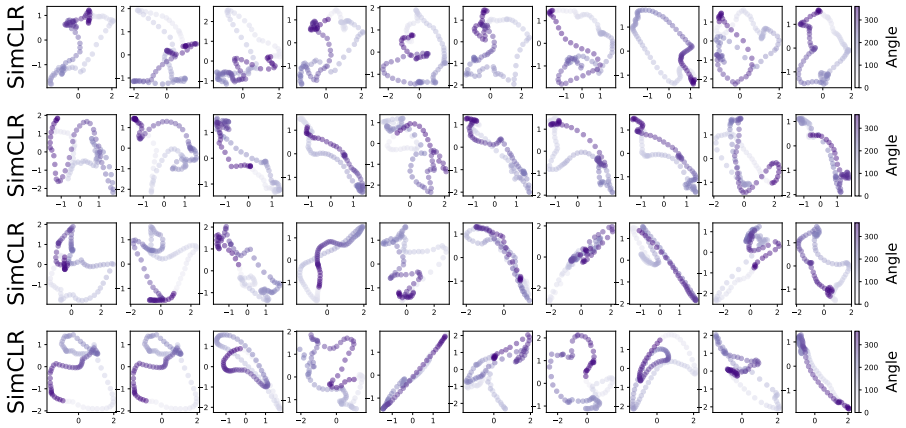

Figure 16: Additional trajectories through the embedding space of a DeepSet trained with SimCLR.

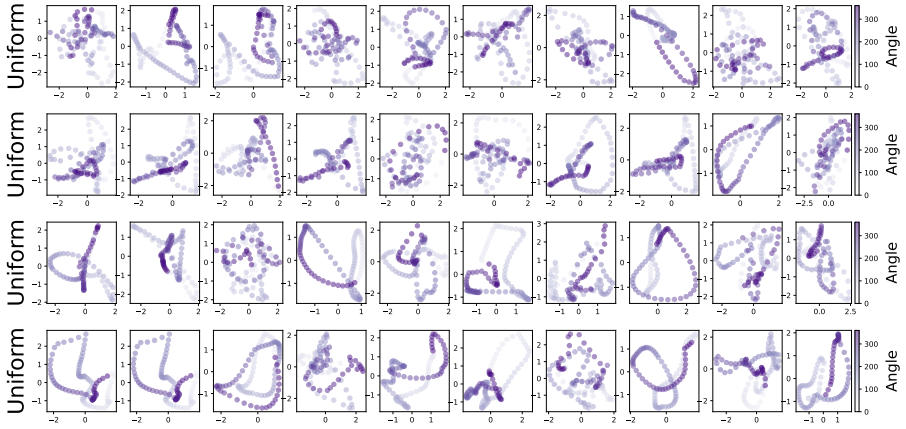

Figure 17: Additional trajectories through the embedding space of a DeepSet trained with the uniformity loss $\mathcal{L}_{\text{unif}}$.

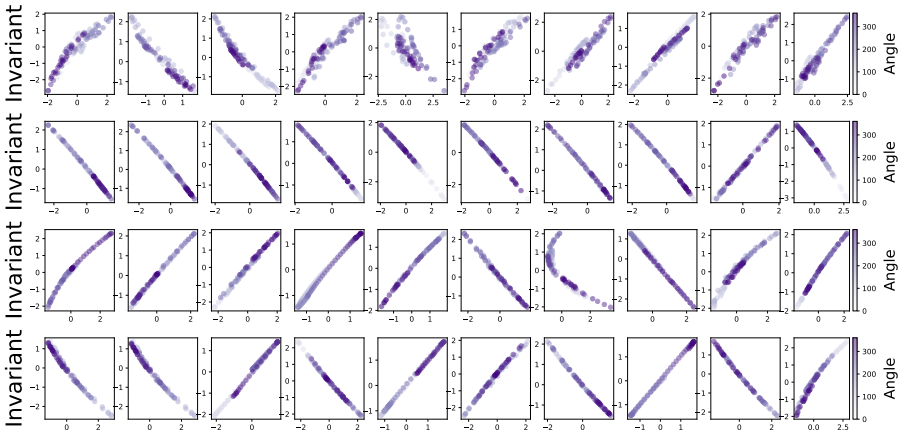

Figure 18: Additional trajectories through the embedding space of a DeepSet trained with the invariance loss $\mathcal{L}_{\text{inv}}$.

G Discussion

Converting transformations that are complex in input space into simple transformations in embedding space has many potential uses. For instance, modifying data (e.g., in order to reason about counterfactuals) can be viewed as transforming one embedding to another. If the sought after transformation was *simple* and *predictable*, it may be easier to find. Similarly, generalizing out-of-distribution is easier when extrapolating linearly [Xu et al., 2021], suggesting that linear transformations of embedding space may facilitate more reliable generalization. This work considers several design principles that may be broadly relevant: 1) *learned* equivariance preserves the expressivity of backbone architectures, and in some cases may be easier for model design than hard-coded equivariance, 2) linear group actions are desirable, but require carefully designed objectives (similar in spirit to the principle of *parsimony* [Ma et al., 2022], also advocated for by Shakerinava et al. [2022]), and 3) orthogonal (and related) symmetries are a promising structure for Siamese network training as they can be efficiently learned using *pair-wise* data comparisons.

Limitations. While our method, CARE, learns embedding spaces with many advantages over prior contrastive learning embedding spaces, there are certain limitations that we acknowledge here. First, we do not provide a means to directly identify the rotation corresponding to a specific transformation. Instead, our approach allows the recovery of the rotation by solving Wahba's problem. However, this requires solving an instance of Wahba's for each augmentation of interest. Future improvements that develop techniques for quickly and easily (i.e., without needing to solve an optimization problem) identifying specific rotations would be a valuable improvement, enhancing the steerability of our models. Second, it is worth noting that equivariant contrastive methods, including CARE, only achieve approximate equivariance. This is a fundamental challenge shared by all such methods, as it is unclear how to precisely encode exact equivariance. The question remains open as to a) whether this approximate equivariance should be considered damaging in the first place, and if so, b) whether scaling techniques can sufficiently produce reliable approximate equivariance to enable the diverse applications that equivariance promises. Addressing this challenge is a crucial area for future research and exploration in the field. Each of these limitations points to valuable directions for future work.

Broader impact. Through our self-supervised learning method CARE we explore foundational questions regarding the structure and nature of neural network representation spaces. Currently, our approaches are exploratory and not ready for integration into deployed systems. However, this line of work studies self-supervised learning and therefore has the potential to scale and eventually contribute to systems that do interact with humans. In such cases, it is crucial to consider the usual safety and alignment considerations. However, beyond this, CARE, offers insights into algorithmic approaches for controlling and moderating model behavior. Specifically, CARE identifies a simple rotation of embedding space that corresponds to a change in the attribute of the data. In principle, this transformation could be used to "canonicalize" data, preventing the model from relying on certain attributes in decision-making. Additionally, controlled transformations of embeddings could be used to debias model responses and achieve desired variations in output. It is important to note that while our focus is on the core methodology, we do not explore these possibilities in this particular work.