The Hidden Pitfalls of the Cosine Similarity Loss

Andrew Draganov DRAGANOVANDREW@CS.AU.DK
Aarhus University

Sharvaree Vadgama S.P.VADGAMA@UVA.NL
University of Amsterdam

Erik J. Bekkers E.J.BEKKERS@UVA.NL
University of Amsterdam

Abstract

We show that the gradient of the cosine similarity between two points goes to zero in two under-explored settings: (1) if a point has large magnitude or (2) if the points are on opposite ends of the latent space. Counterintuitively, we prove that optimizing the cosine similarity between points forces them to *grow* in magnitude. Thus, (1) is unavoidable in practice. We then observe that these derivations are extremely general – they hold across deep learning architectures and for many of the standard self-supervised learning (SSL) loss functions. This leads us to propose *cut-initialization*: a simple change to network initialization that helps all studied SSL methods converge faster.

1. Introduction

Self-supervised learning (SSL) methods can learn robust, multi-use representations in the absence of labeled data by ensuring that similar (positive) inputs are mapped to nearby locations in embedding space. Despite their success, we find that the properties of the cosine similarity – the common loss function across SSL methods – have gone under-studied.

Indeed, we verify that the cosine similarity loss has unexpected interactions with gradient descent. Namely, the gradient acting on a point is near-zero if the point has large magnitude or if it is on the opposite side of the space from its counterpart. We prove that both cases impose at least a quadratic slowdown on the gradient descent convergence. Furthermore, we show that optimizing the cosine similarity between two points *must* cause those points to grow in magnitude.

This leads to an unfortunate catch-22: SSL methods can only be trained under small embedding norms but optimizing the loss-function grows those same embedding norms. Furthermore, this derivation holds for any objective that is itself a function of the cosine similarity (such as the InfoNCE loss [16]) and is *independent* of the SSL architecture, implying that almost all SSL models are affected. We experimentally verify that this is a concern in practice by showing that large embedding norms indeed slow down SSL convergence across architectures and training paradigms. We therefore propose *cut-initialization* to mitigate the issue and verify that, when paired with ℓ_2-normalization, cut-initialization improves convergence across all settings.

2. Preliminaries and Related Work

SSL representations are learned from a dataset without utilizing any labels. This is typically done by obtaining two augmented variants x_i and x_j from an input image x and ensuring that the cor-

© A. Draganov, S. Vadgama & E.J. Bekkers.

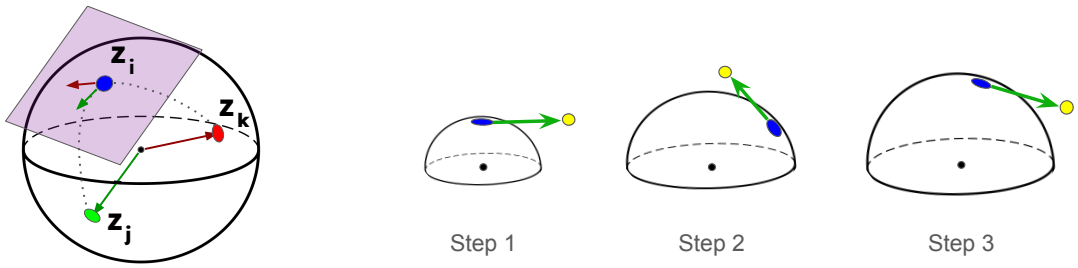

Figure 1: *Left*: The gradients w.r.t. z_i in Proposition 1 and Corollary 4 exclusively exist in the tangent space at $\boldsymbol{z}_i$. *Right*: The growing embeddings in Corollary 2. Blue points represent z_i at iterations $t = 1, 2, 3$. Yellow points represent z_i', i.e. the result of each step of gradient descent.

responding embeddings z_i and z_j have high cosine similarity. We refer to (x_i, x_j) (resp. (z_i, z_j)) as 'positive' pairs of points (resp. embeddings). Each SSL method then introduces an additional mechanism that prevents the entire learned representation from collapsing to a single embedding.

SSL methods. SimCLR [3] follows a line of research of contrastive methods for self-supervised learning [5, 10, 11, 14, 16] which all use repulsions from *negative* samples, z_k, to prevent collapse. These negative samples are the other samples in the batch, implying that no two inputs can be mapped to the same location. Thus, SimCLR minimizes the InfoNCE loss [16] for embedding z_i:

$$\mathcal{L}_i(\mathbf{Z}) = -\log \frac{\text{ExpSim}(z_i, z_j)}{\sum_{k \not= i} \text{ExpSim}(z_i, z_k)} = \underbrace{-\hat{z}_i^\top \hat{z}_j}_{\mathcal{L}_i^{\mathcal{A}}(\mathbf{Z})} + \underbrace{\log\left(S_i\right)}_{\mathcal{L}_i^{\mathcal{R}}(\mathbf{Z})}, \tag{1}$$

where $\hat{z}$ is the unit vector of z, $\text{ExpSim}(a, b) = \exp\left(\hat{a}^\top \hat{b}\right)$ is the exponent of the cosine similarity, and $S_i = \sum_{k \not= i} \text{ExpSim}(z_i, z_k)$ is the sum over the repulsive terms. For clarity, we split the loss term into attraction $\mathcal{L}_i^{\mathcal{A}}(\mathbf{Z})$ and repulsion $\mathcal{L}_i^{\mathcal{R}}(\mathbf{Z})$ loss functions[1], where $\mathcal{L}_i^{\mathcal{A}}(\mathbf{Z})$ is the negative cosine similarity between z_i and z_j. Surprisingly, *BYOL* [8] and *SimSiam* [4] showed that one can optimize $\mathcal{L}_i^{\mathcal{A}}$, the cosine similarity between positive pairs, and avoid collapse by only applying the gradients to embedding z_i (rather than to both z_i and z_j). We refer to these as *non-contrastive* methods.

Across SSL methods, the quality of a learned representation is measured by training classifiers on the learned representations. If the classifier achieves high accuracy then embeddings corresponding to the same class must be near one another. As is standard, we use the k-nn classifier accuracy (this effectively lower-bounds the linear classifier's finetuning accuracy [17]).

SSL Learning Dynamics. The seminal work of [25] showed that repulsion-based contrastive learning must satisfy two requirements: all positive pairs must be near one another (alignment) and all negative samples must spread evenly over the hypersphere (uniformity). They then argued that these requirements lead to classes grouping together. Expanding on this blueprint, subsequent works have sought to formalize the learning capacity of contrastive methods [9, 13, 21, 25, 29] while much of the research studying non-contrastive methods has focused on how their architectural components help to prevent collapse [1, 7, 12, 20, 23, 26].

1. The result of optimizing the attraction term is that the positive pair of embeddings is pulled closer together. Similarly, optimizing the repulsion term causes the negative embedding pairs to be pushed apart from each other.

Importantly, both the contrastive and non-contrastive literature assumes that an embedding's direction in latent space is its relevant parameter. The magnitude's role has, to our knowledge, only been considered in [24] and [27]. The former shows that replacing the softmax's inner products with the cosine similarity resolves distributional issues in the softmax function. The latter identifies that embedding norms affect the cosine similarity but resolves this by simply standardizing the embeddings to have consistent norms. Our work shows that the embedding norms will always grow when optimizing the cosine similarity. Thus, we show that standardizing the norms is insufficient – they must also be minimized. We lastly note that it is an open question why SSL models require ℓ_2-normalization [8, 18, 23] in order to converge. Our work resolves this question.

3. The Cosine Similarity's Dynamics

We begin with the gradients of the cosine similarity loss with respect to an arbitrary point z_i. Throughout this section, we refer to $\mathbf{Z}$ as any set of points in $\mathbb{R}^d$, with no other assumptions over the distribution.

Proposition 1 (Prop. 3 in [27]; proof in A.1) *Let z_i and z_j be two points in $\mathbb{R}^d$ and define $\mathcal{L}_i^{\mathcal{A}}(\mathbf{Z}) = -\hat{z}_i^\top \hat{z}_j$. Let ϕ_{ij} be the angle between z_i and z_j. Then the gradient of $\mathcal{L}_i^{\mathcal{A}}(\mathbf{Z})$ w.r.t. z_i is*

$$\nabla_i^{\mathcal{A}} = -\frac{1}{||z_i||}\left(\mathbf{I} - \frac{z_i z_i^\top}{||z_i||^2}\right)\frac{\mathbf{z}_j}{||z_j||} = -\left(\frac{\hat{z}_j}{||z_i||}\right)_{\perp z_i}$$

where $a_{\perp b}$ is the component of a orthogonal to b. This has magnitude $||\nabla_i^{\mathcal{A}}|| = \frac{\sin(\phi_{ij})}{||z_i||}$.

This has an easy interpretation: $\mathbf{I} - \frac{z_i z_i^\top}{||z_i||^2}$ projects the unit vector $\hat{z}_j$ onto the subspace orthogonal to $\boldsymbol{z}_i$ and it is then inversely scaled by $||z_i||$. We visualize this in Figure 1 (left), where the purple plane represents the projection onto the tangent space at z_i. Corollary 4 in Section B of the Appendix shows that a similar behavior holds for the InfoNCE loss function. As a consequence of these results, we see that optimizing the cosine similarity or the InfoNCE loss can *only grow the embeddings*:

Corollary 2 (Proof in A.2) *Let z_i and z_j be positive embeddings with angle ϕ_{ij}. Let z_i' be the embedding after applying the gradients in Proposition 1 or Corollary 4 to z_i via a step of gradient descent. Then $||z_i'|| \geq ||z_i||$.*

We consider this surprising: one would expect that optimizing the cosine similarity would bring points "in" rather than "out". Nonetheless, the results in Prop. 1 and Corollary 2 reveal an inevitable catch-22: we require small embeddings to optimize the cosine similarity but optimizing the cosine similarity grows the embeddings. This is the key dynamic that [27] missed. Note, these results do not depend on the embedding pipeline – they simply assume that we apply the cosine similarity onto some distribution $\mathbf{Z}$. Furthermore, these results hold for the mean squared error between normalized embeddings, since $||\hat{z}_i - \hat{z}_j||_2^2 = 2 - 2\hat{z}_i^\top \hat{z}_j$. We visualize Corollary 2 in Figure 1 (right).

Effects on Convergence. Our key insights lie in the cosine similarity's (and, by extension, the InfoNCE loss's) gradient magnitude. Namely, it goes to zero when an embedding has large norm or has a large angle to its positive-pair counterpart. We refer to these as the *embedding-norm* and *opposite-halves* effects, respectively. We note that both are counter-intuitive: the former contradicts SSL intuition that an embedding's direction is the relevant term [25] and the latter implies that the maximal loss incurs the smallest gradients. Despite this, both quadratically slow down convergence:

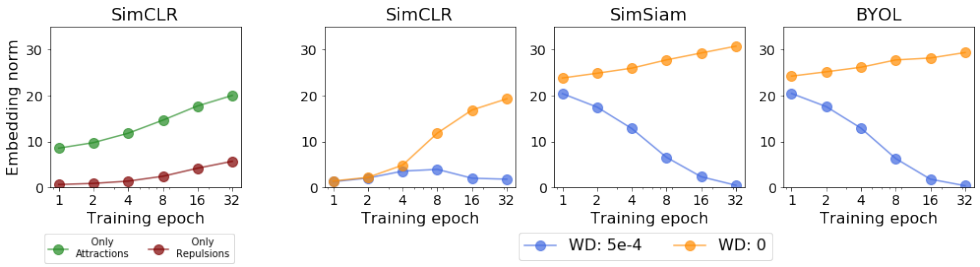

Figure 3: *Left*: The embedding norms when optimizing the subsets of the InfoNCE loss function for SimCLR. *Right*: The mean embedding norms for SimCLR/SimSiam/BYOL as a function of the weight-decay. Note, we use the terms ℓ_2-normalization and weight-decay interchangeably.

Theorem 3 (Proof in A.3) *Let z_i and z_j be embeddings with equal norm, i.e. $||z_i|| = ||z_j|| = \rho$. Let $z_i' = z_i + \frac{\gamma}{\rho}(z_j)_{\perp z_i}$ and $z_j' = z_j + \frac{\gamma}{\rho}(z_i)_{\perp z_j}$ be the embeddings after maximizing the cosine similarity via a step of gradient descent with learning rate γ. Then the change in cosine similarity is bounded from above by:*

$$\hat{z}_i'^\top \hat{z}_j' - \hat{z}_i^\top \hat{z}_j < \frac{2\gamma \sin^2 \phi_{ij}}{\rho^2}. \tag{2}$$

To visualize this slowdown, we apply the cosine similarity gradients to sets of random samples in $\mathbb{R}^{20}$. By varying the mean embedding norms and ϕ_{ij} values, we can then observe how much each parameter affects the convergence. Namely, we apply the gradients in Prop. 1 until the mean cosine similarity exceeds 0.999. We plot the number of steps until convergence in Figure 2. There, we see that although both effects incur quadratic slowdowns, having large embedding norms is *significantly* worse for optimization ($\sim$100$\times$ slowdown) than having large angles between positive pairs ($\sim$2$\times$ slowdown).

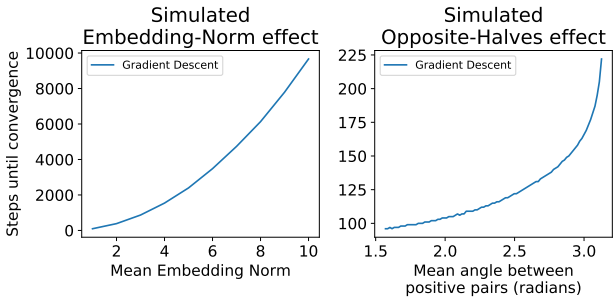

Figure 2: The effect of the embedding norm and angle between positive samples on the convergence rate.

We now discuss whether these effects occur in practice. First, we plot the embedding norms during training on Cifar10 in Figure 3 and see that, across SSL models, the norms indeed grow during training. Thus, Corollary 2 is experimentally supported. Furthermore, applying ℓ_2-normalization counteracts this growth since it forces the embeddings to shrink over time. Consequently, Table 1 shows that ℓ_2-normalization directly affects SSL model quality – small embeddings lead to higher k-nn accuracies. We lastly plot the rate at which positive pairs have angle greater than $\frac{\pi}{2}$ in Table 2 and see that, although positive pairs may have large angle at epoch 1, by epoch 16 this effectively stops occurring. We conclude that the embedding-norm effect indeed slows down SSL training while the opposite-halves effect has a negligible impact (further evidence in Appendix C.1).

		Cifar10		Cifar100			Imagenet-100			Modified Flowers		
		Epoch		Epoch			Epoch			Epoch		
		100	500	100	300	500	100	300	500	100	300	500
SimCLR*	Default	75.7	85.2	33.9	48.1	51.9	44.6	55.9	59.4	18.9	28.3	35.8
	Cut ($c = 3$)	79.6	86.1	42.3	50.8	52.6	46.7	58.1	60.9	23.7	50.8	65.3
SimSiam	Default	67.1	87.0	23.7	50.1	58.6	33.6	61.1	62.0	15.3	16.5	16.9
	Cut ($c = 9$)	77.7	89.0	41.1	59.4	61.7	52.9	63.7	67.2	15.9	27.6	38.5
BYOL	Default	69.5	88.2	18.8	53.5	61.9	26.0	65.8	67.7	16.1	17.4	17.1
	Cut ($c = 9$)	82.2	88.6	46.2	62.4	61.4	54.2	68.7	68.6	15.9	27.6	43.1

Table 3: k-nn accuracies for default and cut-initialized training.

Weight Decay	SimCLR		SimSiam		BYOL	
	k-nn	$\|\|z_i\|\|$	k-nn	$\|\|z_i\|\|$	k-nn	$\|\|z_i\|\|$
$\lambda =$ 5e-2	10.0	0.0	10.0	0.0	10.0	0.0
$\lambda =$ 5e-3	54.0	0.10	57.2	0.47	**78.0**	0.45
$\lambda =$ 5e-4	**81.3**	0.41	**73.3**	0.81	74.3	0.70
$\lambda =$ 5e-6	75.7	13.5	48.2	24.5	47.8	23.2
$\lambda = 0$	75.3	16.4	47.9	31.4	47.3	30.1

Table 1: Effect of ℓ_2-normalization on SSL training for Cifar10. We report the k-nn accuracy and the embedding norms for various weight decay parameters and SSL models. $||z_i||$ represents the mean embedding norm.

Model	Dataset	Effect Rate Epoch 1	Effect Rate Epoch 16
SimCLR	Imagenet-100	2%	0%
	Cifar-100	11%	1%
SimSiam	Imagenet-100	26%	1%
	Cifar-100	21%	0%
BYOL	Imagenet-100	28%	1%
	Cifar-100	20%	0%

Table 2: The rate at which embeddings are on opposite sides of the latent space for various datasets and SSL models.

4. Cut-Initialization

The previous sections showed that the embedding-norm effect occurs in SSL models and that weight decay eventually remedies it. Rather than waiting for the weight-decay to slowly shrink the embeddings, we propose to initialize the embeddings directly at the correct size via *cut-initialization*: an initialization scheme where we divide each layer's weights with cut-constant $c > 1$ before training.

We refer to Tables 5 and 6 to study the relationship between the cut-constant and the weight-decay. Specifically, we analyze the classification accuracy after 100 epochs and see that, in both the contrastive and non-contrastive settings, pairing cut-initialization with weight decay accelerates the training process. Based on these results, we default to cut-constant $c = 3$ for contrastive methods and $c = 9$ for non-contrastive ones. Table 3 then evaluates the effectiveness of cut-initialization on full training runs[2]. The takeaway is that cut-initialization essentially guarantees faster convergence.

We note that the original Flowers102 dataset [15] is nearly impossible for self-supervised image recognition. We therefore create a new training split using the supplementary imbalanced class examples. Each class in our *modified-flowers* dataset has between 50 and 250 samples, implying a long-tailed data distribution [28]. Despite this still being challenging for the default models, incorporating cut-initialization allows them to obtain significantly higher accuracies in Table 3. Thus, not only does cut-initialization accelerate training, it can sometimes facilitate it entirely.

It remains to show that cut-initialization works across SSL models and backbone architectures. To this end, Table 3 shows that cut-initialization still provides the expected boost in performance for MoCov2 and for MoCov3 (which uses a ViT-s trasnformer backbone). We point out that Dino's objective function does not depend on the cosine similarity [2]. Consequently, it does not (and

2. Due to lack of compute, we use a batch-size of 128 with the Resnet50 backbone, indicated by the asterisk.

should not) benefit from cut-initialization. This supports our analysis: using cut-initialization in theoretically unmotivated settings no longer has a positive impact on training.

		Top-1	Top-5
MoCo V2	$c = 1$	38.7	69.7
	$c = 3$	43.8	71.8
MoCo V3	$c = 1$	54.8	82.7
	$c = 3$	58.8	86.0
DINO	$c = 1$	43.7	76.6
	$c = 3$	29.0	55.6

Table 4: Imagenet-100 k-nn accuracy (epoch 100) for MoCo and Dino.

SimCLR

		Weight Decay			
		1e-8	<u>1e-6</u>	5e-6	1e-5
Cut	$c = 1$	40.8	40.5	40.9	41.5
	$c = 2$	42.7	42.8	42.9	42.2
	$c = 4$	42.3	41.4	42.0	41.1
	$c = 8$	37.1	36.8	37.9	37.3

Table 5: SimCLR k-nn accuracy (epoch 100) on Cifar100 for various cut-constants and weight-decays. Default weight-decay is underlined.

SimSiam

		Weight Decay			
		5e-5	1e-4	<u>5e-4</u>	1e-3
Cut	$c = 1$	36.7	38.5	40.5	44.7
	$c = 2$	41.1	44.0	48.8	42.1
	$c = 4$	40.2	41.2	49.8	49.1
	$c = 8$	44.4	46.4	50.2	53.2

Table 6: SimSiam k-nn accuracy (epoch 100) on Imagenet-100 for various values of c and λ. Default weight-decay is underlined.

References

[1] Randall Balestriero and Yann LeCun. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. *Advances in Neural Information Processing Systems*, 35:26671–26685, 2022.

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[4] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[6] Abe Fetterman and Josh Albrecht. Understanding self-supervised and contrastive learning with bootstrap your own latent (byol). *Untitled AI, August*, 2020.

[7] Quentin Garrido, Yubei Chen, Adrien Bardes, Laurent Najman, and Yann Lecun. On the duality between contrastive and non-contrastive self-supervised learning. *arXiv preprint arXiv:2206.02574*, 2022.

[8] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[9] Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34:5000–5011, 2021.

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[11] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International conference on machine learning*, pages 4182–4192. PMLR, 2020.

[12] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

[13] Yeskendir Koishekenov, Sharvaree Vadgama, Riccardo Valperga, and Erik J Bekkers. Geometric contrastive learning. In *4th Visual Inductive Priors for Data-Efficient Deep Learning Workshop*, 2023. URL https://openreview.net/forum?id=cE4BY5XrzR.

[14] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6707–6717, 2020.

[15] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[16] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[17] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[18] Reviewer-kELi. Openreview comment. https://openreview.net/forum?id=rw2dkPZOoQ&noteId=kFoyEsqLRv. Accessed: 2024-03-04.

[19] Pierre H Richemond, Jean-Bastien Grill, Florent Altché, Corentin Tallec, Florian Strub, Andrew Brock, Samuel Smith, Soham De, Razvan Pascanu, Bilal Piot, et al. Byol works even without batch statistics. *arXiv preprint arXiv:2010.10241*, 2020.

[20] Pierre Harvey Richemond, Allison Tam, Yunhao Tang, Florian Strub, Bilal Piot, and Felix Hill. The edge of orthogonality: a simple view of what makes byol tick. In *International Conference on Machine Learning*, pages 29063–29081. PMLR, 2023.

[21] Nikunj Saunshi, Jordan Ash, Surbhi Goel, Dipendra Misra, Cyril Zhang, Sanjeev Arora, Sham Kakade, and Akshay Krishnamurthy. Understanding contrastive learning requires incorporating inductive biases. In *International Conference on Machine Learning*, pages 19250–19286. PMLR, 2022.

[22] Ian Smith, Janosch Ortmann, Farnoosh Abbas-Aghababazadeh, Petr Smirnov, and Benjamin Haibe-Kains. On the distribution of cosine similarity with application to biology. *arXiv preprint arXiv:2310.13994*, 2023.

[23] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pages 10268–10278. PMLR, 2021.

[24] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, MM '17. ACM, October 2017. doi: 10.1145/3123266.3123359. URL http://dx.doi.org/10.1145/3123266.3123359.

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

[26] Chaoning Zhang, Kang Zhang, Chenshuang Zhang, Trung X Pham, Chang D Yoo, and In So Kweon. How does simsiam avoid collapse without negative samples? a unified understanding with self-supervised contrastive learning. *arXiv preprint arXiv:2203.16262*, 2022.

[27] Dingyi Zhang, Yingming Li, and Zhongfei Zhang. Deep metric learning with spherical embedding. *Advances in Neural Information Processing Systems*, 33:18772–18783, 2020.

[28] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[29] Roland S Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In *International Conference on Machine Learning*, pages 12979–12990. PMLR, 2021.

Appendix A. Proofs

A.1. Proof of Proposition 1

Proof We are taking the gradient of $\mathcal{L}_i^{\mathcal{A}}$ as a function of z_i. The principal idea is that the gradient has a term with direction $\hat{z}_j$ and a term with direction $-\hat{z}_i$. We then disassemble the vector with direction $\hat{z}_j$ into its component parallel to z_i and its component orthogonal to z_i. In doing so, we find that the two terms with direction z_i cancel, leaving only the one with direction orthogonal to z_i.

Writing it out fully, we have $\mathcal{L}_i^{\mathcal{A}} = -z_i^\top z_j/(||z_i|| \cdot ||z_j||)$. Taking the gradient amounts to using the quotient rule, with $f = -z_i^\top z_j$ and $g = ||z_i|| \cdot ||z_j|| = \sqrt{z_i^\top z_i} \cdot \sqrt{z_j^\top z_j}$. Taking the derivative of each, we have

$$
\begin{aligned}
f' &= -\mathbf{z}_j \\
g' &= ||z_j|| \frac{z_i}{\sqrt{z_i^\top z_i}} = ||z_j|| \frac{\mathbf{z}_i}{||z_i||} \\
\implies \frac{f'g - g'f}{g^2} &= \frac{-\left(\mathbf{z}_j \cdot ||z_i|| \cdot ||z_j||\right) + \left(||z_j|| \frac{\mathbf{z}_i}{||z_i||} \cdot z_i^\top z_j\right)}{||z_i||^2 \cdot ||z_j||^2} \\
&= \frac{-\mathbf{z}_j}{||z_i|| \cdot ||z_j||} + \frac{\mathbf{z}_i z_i^\top z_j}{||z_i||^3 ||z_j||},
\end{aligned}
$$

where we use boldface $\mathbf{z}$ to emphasize which direction each term acts along. We now substitute $\cos(\phi_{ij}) = z_i^\top z_j/(||z_i|| \cdot ||z_j||)$ in the second term to get

$$
\frac{f'g - g'f}{g^2} = \frac{-\hat{z}_j}{||z_i||} + \frac{\mathbf{z}_i \cos(\phi)}{||z_i||^2} \tag{3}
$$

It remains to separate the first term into its sine and cosine components and perform the resulting cancellations. To do this, we take the projection of $\hat{z}_j = \mathbf{z}_j/||z_j||$ onto $\mathbf{z}_i$ and onto the plane orthogonal to $\mathbf{z}_i$. The projection of $\hat{z}_j$ onto $\mathbf{z}_i$ is given by

$$
\cos \phi_{ij} \frac{\mathbf{z}_i}{||z_i||}
$$

while the projection of $\mathbf{z}_j/||z_j||$ onto the plane orthogonal to $\mathbf{z}_i$ is

$$
\left(\mathbf{I} - \frac{z_i z_i^\top}{||z_i||^2}\right) \frac{\mathbf{z}_j}{||z_j||}.
$$

It is easy to assert that these components sum to $\mathbf{z}_j/||z_j||$ by replacing the $\cos \phi_{ij}$ by $\frac{z_i^\top z_j}{||z_i|| \cdot ||z_j||}$.

We plug these into Eq. 3 and cancel the first and third term to arrive at the desired value:

$$\begin{aligned}\frac{f'g - g'f}{g^2} =& -\frac{1}{||z_i||}\cos\phi\frac{\mathbf{z}_i}{||z_i||} \\ &- \frac{1}{||z_i||}\cdot\left(\mathbf{I} - \frac{z_i z_i^\top}{||z_i||^2}\right)\frac{\mathbf{z}_j}{||z_j||} \\ &+ \frac{\mathbf{z}_i\cos(\phi)}{||z_i||^2} \\ =& \frac{-1}{||z_i||}\cdot\left(\mathbf{I} - \frac{z_i z_i^\top}{||z_i||^2}\right)\frac{\mathbf{z}_j}{||z_j||}.\end{aligned}$$

■

A.2. Proof of Corollary 2

Proof First, consider that we applied the cosine similarity's gradients from Proposition 1. Since z_i and $(z_j)_{\perp z_i}$ are orthogonal, $||z_i'||_2^2 = ||z_i||^2 + \frac{\gamma^2}{||z_i||^2}||(z_j)_{\perp z_i}||^2$. The second term is positive if $\sin\phi_{ij} > 0$. Figure 1 visualizes this.

The same exact argument holds for the InfoNCE gradients. The gradient is orthogonal to the embedding, so a step of gradient descent can only increase the embedding's magnitude. ■

A.3. Proof of Theorem 3

We first restate the theorem:

Let z_i and z_j be positive embeddings with equal norm, i.e. $||z_i|| = ||z_j|| = \rho$. Let z_i' and z_j' be the embeddings after 1 step of gradient descent with learning rate γ. Then the change in cosine similarity is bounded from above by:

$$\hat{z}_i'^\top\hat{z}_j' - \hat{z}_i^\top\hat{z}_j < \frac{\gamma\sin^2\phi_{ij}}{\rho^2}\left[2 - \frac{\gamma\cos\phi}{\rho^2}\right].$$

We now proceed to the proof:

Proof Let z_i and z_j be two embeddings with equal norm[3], i.e. $||z_i|| = ||z_j|| = \rho$. We then perform a step of gradient descent to maximize $\hat{z}_i^\top\hat{z}_j$. That is, using the gradients in 1 and learning rate γ, we obtain new embeddings $z_i' = z_i + \frac{\gamma}{||z_i||}(\hat{z}_j)_{\perp z_i}$ and $z_j' = z_j + \frac{\gamma}{||z_j||}(\hat{z}_i)_{\perp z_j}$. Going forward, we write $\delta_{ij} = (\hat{z}_j)_{\perp z_i}$ and $\delta_{ji} = (\hat{z}_i)_{\perp z_j}$, so $z_i' = z_i + \frac{\gamma}{\rho}\delta_{ij}$ and $z_j' = z_j + \frac{\gamma}{\rho}\delta_{ji}$. Notice that since z_i and δ_{ij} are orthogonal, by the Pythagorean theorem we have $||z_i'||^2 = ||z_i||^2 + \frac{\gamma^2}{\rho^2}||\delta_{ij}||^2 \geq ||z_i||^2$. Lastly, we define $\rho' = ||z_i'|| = ||z_j'||$.

We are interested in analyzing $\hat{z}_i'^\top\hat{z}_j' - \hat{z}_i^\top\hat{z}_j$. To this end, we begin by re-framing $\hat{z}_i'^\top\hat{z}_j'$:

$$\begin{aligned}\hat{z}_i'^\top\hat{z}_j' &= \left(\frac{z_i + \frac{\gamma}{\rho}\delta_{ij}}{\rho'}\right)^\top\left(\frac{z_j + \frac{\gamma}{\rho}\delta_{ji}}{\rho'}\right) \\ &= \frac{1}{\rho'^2}\left[z_i^\top z_j + \gamma\frac{z_i^\top\delta_{ji}}{\rho'} + \gamma\frac{z_j^\top\delta_{ij}}{\rho'} + \gamma^2\frac{\delta_{ij}^\top\delta_{ji}}{\rho'^2}\right].\end{aligned}$$

3. We assume the Euclidean distance for all calculations.

We now consider that, since δ_{ij} is the projection of $\hat{z}_j$ onto the subspace orthogonal to z_i, we have that the angle between z_i and δ_{ji} is $\pi/2 - \phi_{ij}$. Plugging this in and simplifying, we obtain

$$\begin{aligned} z_i^\top \delta_{ji} &= ||z_i|| \cdot ||\delta_{ji}|| \cos(\pi/2 - \phi_{ij}) \\ &= ||z_i|| \cdot ||\delta_{ji}|| \sin \phi_{ij} \\ &= \rho \sin^2 \phi_{ij}. \end{aligned}$$

By symmetry, the same must hold for $z_j^\top \delta_{ij}$.

Similarly, we notice that the angle ψ_{ij} between δ_{ij} and δ_{ji} is $\psi_{ij} = \pi - \phi_{ij}$. The reason for this is that we must have a quadrilateral whose four internal angles must sum to 2π, i.e. $\psi_{ij} + \phi_{ij} + 2\frac{\pi}{2} = 2\pi$. Thus, we obtain $\delta_{ij}^\top \delta_{ji} = ||\delta_{ij}|| \cdot ||\delta_{ji}|| \cos(\psi) = -\sin^2 \phi_{ij} \cos \phi_{ij}$.

We plug these back into our equation for $\hat{z}_i'^\top \hat{z}_j'$ and simplify:

$$\begin{aligned} \hat{z}_i'^\top \hat{z}_j' &= \frac{1}{\rho'^2} \left[z_i^\top z_j + \gamma \frac{z_i^\top \delta_{ji}}{\rho} + \gamma \frac{z_j^\top \delta_{ij}}{\rho} + \gamma^2 \frac{\delta_{ij}^\top \delta_{ji}}{\rho^2} \right] \\ &= \frac{1}{\rho'^2} \left[z_i^\top z_j + \gamma \frac{\rho \sin^2 \phi_{ij}}{\rho} + \gamma \frac{\rho \sin^2 \phi_{ij}}{\rho} - \gamma^2 \frac{\sin^2 \phi_{ij} \cos \phi_{ij}}{\rho^2} \right] \\ &= \frac{1}{\rho'^2} \left[z_i^\top z_j + 2\gamma \sin^2 \phi_{ij} - \gamma^2 \frac{\sin^2 \phi_{ij} \cos \phi_{ij}}{\rho^2} \right]. \end{aligned}$$

We now consider the original term in question:

$$\begin{aligned} \hat{z}_i'^\top \hat{z}_j' - \hat{z}_i^\top \hat{z}_j &= \frac{1}{\rho'^2} \left[z_i^\top z_j + 2\gamma \sin^2 \phi_{ij} - \gamma^2 \frac{\sin^2 \phi_{ij} \cos \phi_{ij}}{\rho^2} \right] - \frac{z_i^\top z_j}{\rho^2} \\ &\leq \frac{1}{\rho^2} \left[z_i^\top z_j + 2\gamma \sin^2 \phi_{ij} - \gamma^2 \frac{\sin^2 \phi_{ij} \cos \phi_{ij}}{\rho^2} \right] - \frac{z_i^\top z_j}{\rho^2} \\ &= \frac{1}{\rho^2} \left[2\gamma \sin^2 \phi_{ij} - \gamma^2 \frac{\sin^2 \phi_{ij} \cos \phi_{ij}}{\rho^2} \right] \\ &= \frac{\gamma \sin^2 \phi_{ij}}{\rho^2} \left[2 - \frac{\gamma \cos \phi_{ij}}{\rho^2} \right] \\ &\leq \frac{2\gamma \sin^2 \phi_{ij}}{\rho^2} \end{aligned}$$

This concludes the proof. ■

Appendix B. InfoNCE Gradients

The InfoNCE gradient has a similar structure to the cosine similarity's:

Corollary 4 *Let* $\mathbf{Z}$ *be a set of* b *embeddings in* $\mathbb{R}^d$*, with* z_i *corresponding to the* i*-th row of* $\mathbf{Z}$*. Let* $\mathcal{L}_i(\mathbf{Z}) = \mathcal{L}_i^{\mathcal{A}}(\mathbf{Z}) + \mathcal{L}_i^{\mathcal{R}}(\mathbf{Z})$*. Then the gradient of* $\mathcal{L}_i(\mathbf{Z})$ *w.r.t.* z_i *is*

$$\nabla_i^{InfoNCE} = -\frac{1}{||z_i||} \left[(\hat{z}_j)_{\perp z_i} + \sum_{k \not\sim i} \left(\hat{z}_k \cdot \frac{\text{ExpSim}(z_i, z_k)}{S_i} \right)_{\perp z_i} \right]. \tag{4}$$

Proof We are interested in the gradient of $\mathcal{L}_i^{\mathcal{R}}$ with respect to z_i. By the chain rule, we get

$$\begin{aligned}\nabla_i^{\mathcal{R}} &= -\frac{\sum_{k \not= i} \text{ExpSim}(z_i, z_k) \frac{\partial \frac{z_i^\top z_k}{||z_i|| \cdot ||z_k||}}{\partial z_i}}{\sum_{k \not= i} \text{ExpSim}(z_i, z_k)} \\ &= -\frac{\sum_{k \not= i} \text{ExpSim}(z_i, z_k) \frac{\partial \frac{z_i^\top z_k}{||z_i|| \cdot ||z_k||}}{\partial z_i}}{S_i}\end{aligned}$$

It remains to substitute the result of Prop. 1 for $\partial \frac{z_i^\top z_k}{||z_i|| \cdot ||z_k||} / \partial z_i$.

We sum this this with the gradients of the attractive term to obtain the full InfoNCE gradient, completing the proof. ■

In essence, because the InfoNCE loss is a function of the cosine similarity, the chain rule implies that its gradients have similar properties to the cosine similarity's. Specifically, just like those of $\mathcal{L}_i^{\mathcal{A}}$, the gradients of $\mathcal{L}_i^{\mathcal{R}}$ have the properties that (1) they are inversely scaled by $||z_i||$, (2) they are only defined in z_i's tangent space, and (3) optimizing the repulsions cause the embeddings to grow[4]. Since the InfoNCE loss is the sum of $\mathcal{L}_i^{\mathcal{A}}$ and $\mathcal{L}_i^{\mathcal{R}}$, these properties all extend to the InfoNCE loss as well.

Appendix C. Further Discussion and Experiments

C.1. Opposite-Halves Effects

We first consider how much the opposite-halves effect plays a role in SSL training in theory. Referring back to Figure 2, we see that the effect is most impactful when the angle between positive embeddings is close to π, i.e. $\phi_{ij} > \pi - \varepsilon$ for $\varepsilon \to 0$. The following result shows that this is exceedingly unlikely in high-dimensions:

Proposition 5 *Let $x_i, x_j \sim \mathcal{N}(0, \mathbf{I})$ be d-dimensional, i.i.d. random variables and let $0 < \varepsilon < 1$. Then*

$$\mathbb{P}\left[\hat{x}_i^\top \hat{x}_j \geq 1 - \varepsilon\right] \leq \frac{1}{2d(1-\varepsilon)^2}. \tag{5}$$

Proof By [22], the cosine similarity between two i.i.d. random variables drawn from $\mathcal{N}(0, \mathbf{I})$ has expected value $\mu = 0$ and variance $\sigma^2 = 1/d$, where d is the dimensionality of the space. We therefore plug these into Chebyshev's inequality:

$$\begin{aligned}&\Pr\left[\left|\frac{x_i^\top x_j}{||x_i|| \cdot ||x_j||} - \mu\right| \geq k\sigma\right] \leq \frac{1}{k^2} \\ \to &\Pr\left[\left|\frac{x_i^\top x_j}{||x_i|| \cdot ||x_j||}\right| \geq \frac{k}{\sqrt{d}}\right] \leq \frac{1}{k^2}\end{aligned}$$

4. The primary difference is that the repulsive gradients have a weighted average over all the negative samples z_k. We note that this weighted average over a set of unit vectors leads to the repulsive force being smaller than the attractive one.

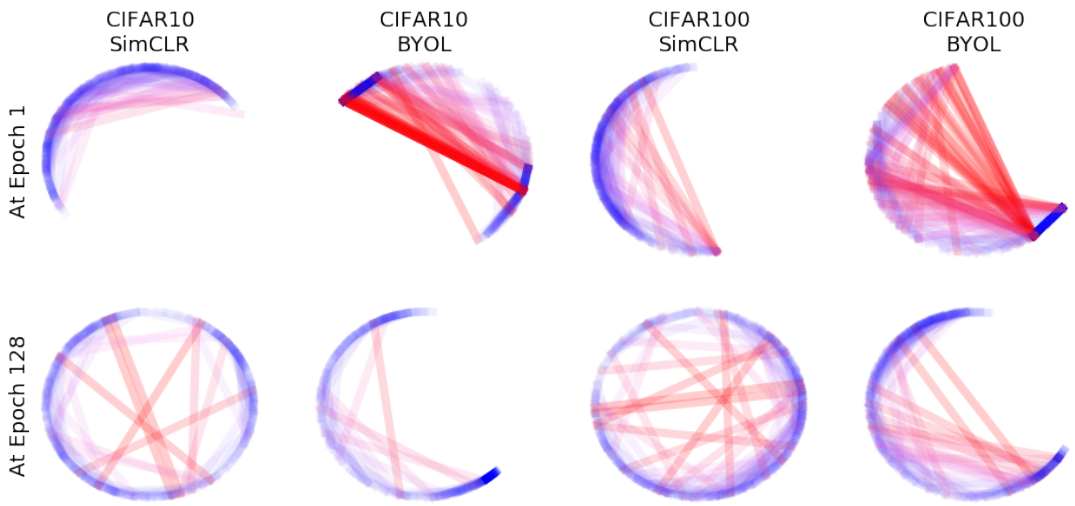

Figure 4: Lines between positive samples in 2D latent space during training. The color goes from red to blue as the cos. sim. goes from -1 to 1.

We now choose $k = \sqrt{d}(1-\varepsilon)$, giving us

$$\mathbb{P}\left[\left|\frac{x_i^\top x_j}{||x_i|| \cdot ||x_j||}\right| \geq 1-\varepsilon\right] \leq \frac{1}{d(1-\varepsilon)^2}.$$

It remains to remove the absolute values around the cosine similarity. Since the cosine similarity is symmetric around 0, the likelihood that its absolute value exceeds $1-\varepsilon$ is twice the likelihood that its value exceeds $1-\varepsilon$, concluding the proof.

We note that this is actually an extremely optimistic bound since we have not taken into account the fact that the maximum of the cosine similarity is 1. ■

Thus, the opposite-halves effect is exceedingly unlikely to occur in theory.

Nonetheless, Table 2 shows that embeddings have angle greater than $\pi/2$ at a rate of 5% and 25% for SimCLR and SimSiam/BYOL, respectively. So even if the 'strongest' variant of the opposite-halves effect is not occurring, a weaker one may still be. To visualize why this happens more for non-contrastive embeddings, we train SimCLR and BYOL with a 2-dimensional projection/prediction latent space (as is done for MoCo in [25]) and draw lines between positive samples in Figure 4. SimCLR performs as expected – it distributes the embeddings across the latent space and keeps positive samples close together. However, Figure 4 suggests that BYOL begins by concentrating its embeddings at antipodal points on the hypersphere and does not spread out evenly from there. Drawing from [4, 6, 19], we attribute this to the batch-normalization that is present in BYOL and SimSiam's projection/prediction heads[5].

Putting these pieces together explains why SimSiam and BYOL have a higher rate of the opposite-halves effect at the start of training. Since they only optimize attractions, they must begin

5. Consider what happens if the distribution has collapsed to a single point right before the batch-normalization layer. Since batch-normalization subtracts the mean and divides by the variance, the single collapsed cluster becomes chaotically distributed across the latent space [6]. Thus, batch-normalized models cannot collapse to a single point.

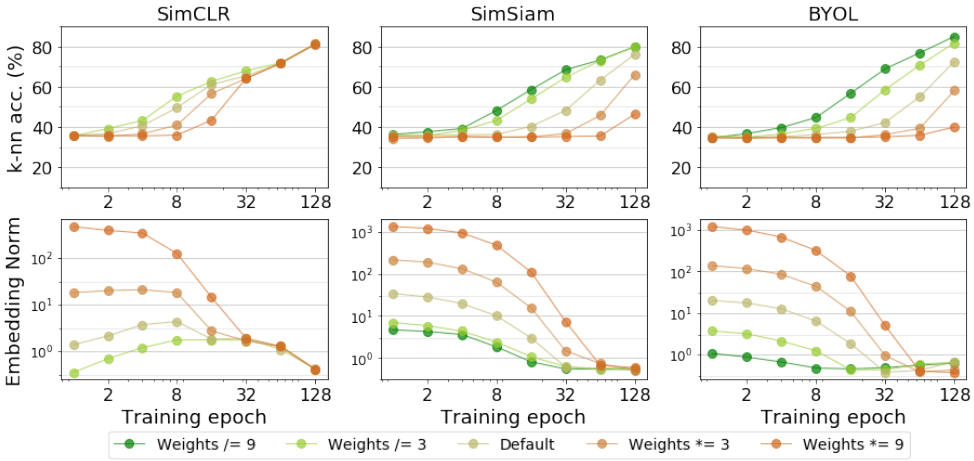

Figure 5: The effect of cut-initialization on Cifar10 SSL representations. x-axis and embedding norm's y-axis are log-scale. $\lambda = 5\text{e}{-4}$ in all experiments.

by concentrating embeddings together. Due to batch-normalization, this concentration occurs in several clusters that straddle the origin. Thus, some positive pairs will necessarily have an angle greater than $\frac{\pi}{2}$ at the beginning of training (seen in the 'Epoch 1' row of Figure 4).

We point out, however, that very early into training (epoch 16), every method has a rate of effectively 0 for the opposite-halves effect. This is visualized in Figure 4, where the learned representations at epoch 128 have very few 'red' lines[6]. Furthermore, the rates in Table **??** measure how often $\phi_{ij} > \frac{\pi}{2}$: the absolute weakest version of the opposite-halves effect. If we instead count the number of times that $\phi_{ij} > \frac{7\pi}{8}$ (the range where the opposite halves effect could start mattering according to Figure 2), then it occurs 0 times across all the experiments in Table **??**. Thus, it is clear that while some weak variant of the opposite-halves effect must occur at the beginning of training, it does not have a strong impact on the convergence dynamics and, in either case, disappears quite quickly.

C.2. Cut-Initialization

To get a preliminary intuition for cut-initialization's impact on the convergence, we plot the effect of the cut constant on the embedding norms and accuracies over training in Figure 5[7]. We see that dividing the network's weights by $c > 1$ leads to immediate convergence improvements in all models. Furthermore, this effect degrades gracefully: as $c > 1$ becomes $c < 1$, the embeddings stay large for longer and, as a result, the convergence is slower. We also see that cut-initialization has a more pronounced effect in attraction-only models – a trend that remains consistent throughout the experiments.

6. Interestingly, Figure 4 also implies that attraction-only methods never reach uniformity over the latent space.
7. To make the effect more apparent, we use weight-decay $\lambda = 5e - 4$ in all models.