TO CONTRAST OR NOT TO CONTRAST: EXPLORING THE STRUCTURE OF CONTRASTIVE LEARNING REPRESENTATIONS

ANIRUDH COWLAGI, ANDY LIU, ARJUN NEERVANNAN

ABSTRACT. A goal of visual representation learning is to produce representations of natural images that reflect our own belief priors about the structure of images as well as the labels given to them. Contrastive learning (CL) has made impressive progress towards the former objective by explicitly encouraging similar examples to have similar representations [1]. Such models can then be readily fine-tuned on a target task – e.g. classification. In the face of this initial success, this work explores various properties of representations learned by self-supervised and supervised contrastive learning methods and verify their contribution to transferability and robustness on downstream tasks. Code required to recreate these experiments can be found at github.com/a-cowlagi/To-Contrast-Or-Not-To-Contrast and github.com/a-cowlagi/SupContrast.

1. INTRODUCTION

Self-supervised contrastive learning introduces human understanding of natural images by forcing random augmentations of an image ("positive" examples) to have similar representations while discriminating on different images [1]. This is done through a two-part contrastive losses: an alignment component and a regularizing component to prevent representation collapse [4]. A classic example is the InfoNCE loss adopted by the popular SimCLR framework [1]. Building on this paradigm are supervised contrastive learning techniques that additionally inject information about the labels into the contrastive loss metric [6].

A particularly relevant characterization is the *geometry* of models trained under self-supervised and supervised CL, compared to traditional supervised learning. We look to examine the shape of the loss landscape on downstream classification tasks. It is fairly well-documented that transferable models tend to locate flatter loss minima and have improved Lipschitzness [9].

To further evaluate the transferability of CL approaches, we consider the *trajectory* taken by CL and supervised pre-training during the fine-tuning phase. We adopt the approach used in [11] and leverage a uniform time coordinate to develop a consistent and meaningful measure of *progress* along a task. We further visualize these trajectories using an isometric embedding of probabilistic models [10].

Finally, we verify the robustness results presented in [5] on semi-supervised CL and supervised CL, which find that CL methods suffer from increased vulnerability to projected gradient descent (PGD) attacks than traditional supervised approaches. We posit that transferability and adversarial robustness go hand in hand – the flatness of the loss landscape to perturbations in the weight space should correspond to flatness of the landscape to perturbations in the input space.

1.1. Contributions.

We make the following contributions in this paper:

- We examine the *transferrability* properties of deep networks trained with contrastive learning methods by considering both the geometry of the loss landscape on downstream tasks *and* the trajectory taken by these models from the end of pretraining. We benchmark these properties by comparing to networks trained using traditional supervised cross-entropy approaches.
- We examine the *robustness* properties of deep networks trained with contrastive learning methods by performing adversarial attacks on the training samples used at the downstream stages and show that contrastive methods are equally, if not more robust, than their traditional supervised cross-entropy counterparts.

2. BACKGROUND

2.1. Details on Contrastive Learning.

Contrastive Learning (CL) methods are self-supervised (or semi-supervised) learning approaches that explicitly encourage models to learn similar latent space representations for similar inputs, and dissimilar representations for different inputs.

In the typical CL process, given a base encoder function $f(\cdot)$, we apply two different transformations on an input datum x to create a *positive pair* of examples x_i and x_j. They are passed through the encoder $f(\cdot)$ to generate latent states

h_i and h_j. A small projection head $g(\cdot)$ is attached to the encoder to map the latent space to the space where the CL loss function is applied.

[1] uses the *InfoNCE* loss function as follows: given two outputs z_i and z_j from the projection head $g()$, we define the similarity between the two vectors to be:

$$\text{sim}(u, v) = \frac{u^T v}{||u||||v||} \tag{1}$$

The loss on a single pair of positive images is then (with the remaining $2(N-1)$ augmented samples in the minibatch deemed as negative examples):

$$\mathcal{L}^{InfoNCE}(z_i, z_j) = -\log \frac{\exp(\text{sim}(z_i, z_j)/T)}{\sum_{k=1,k\neq i}^{2N} \text{sim}(z_i, z_k)/T} \tag{2}$$

Furthermore, when labels are known, they may be leveraged to create new loss functions by allowing positive examples to come from the same class, rather than simply restricting to random augmentations of the anchor. We thus consider the supervised contrastive (SupCon) loss, defined on a single sample i with positive examples $P(i)$ as:

$$\mathcal{L}^{SupCon}(x_i) = -\log \frac{1}{|P(i)|} \sum_{p\in P(i)} \frac{\exp(\text{sim}(z_i, z_p)/T)}{\sum_{k=1,k\neq i}^{2N} \exp \text{sim}(z_i, z_j)/T} \tag{3}$$

2.2. **Transferability and the Geometry of Contrastive Learning Representations.** It has been shown that well-trained transferable models tend to locate flatter loss minima [9]. This makes sense: their weight matrices are likely to stay close to the original (flat) region of pretrained parameters when transfered to a target dataset with similar sturcture. We can verify this fact by visualizing the loss landscape of a model on downtream classification tasks using the approach from [8]. It is interesting to see if this remains the case with contrastive learning methods that utilize entirely different loss functions during the pretraining phase compared to the classification finetuning phase.

In addition to the geometry of the landscape at the *end* of finetuning, we believe the *trajectory* taken by a model from the end of pretraining is equally relevant. We naturally expect that an *efficiently* pretrained model would make "faster" progress towards the ground truth output. Formalizing a notion of progress is challenging because tasks are different, and each model typically is associated with its own time coordinate – its own training progress.

To address this, we adopt the approach from [11] and provide the relevant details below. Suppose we have access to dataset $D = \{(x_n, y_n)\}_{i=1}^N$. We may view a network trained on a classification task on C classes with weights w, as a probabilistic model, P_u on any vector of output labels $\vec{y} \in [1..C]^N$:

$$P_w(\vec{(y)}) = \prod_{i=1}^{N} p_w(y_i|x_i) \tag{4}$$

Let P_0 be the model that simply predicts $p_w(y_n|x_n) = 1/C$ for all samples, and let P^* be the model that predicts the ground truth labels. Any randomly initialized model may expect to follow a trajectory starting from P_0 and moving towards P^*. The *optimal* model follows the shortest such trajectory. Note that for a given sample (x_i, y_i), the vector of the square root of probabilities lies on the surface of a $C-1$ dimensional sphere: $\vec{p_w(y)} = [p_w(y_i = 1|x_i), ..., p_w(y_i = C, x_i)]$. The geodesics on each sample are thus great circles. Any P_w is simply a point on the product manifold of these spheres, so geodesics on these product manifolds are simply the products of the geodesics on each sphere, so we may write the geodesic between two models, P_u, P_v as a function of $\lambda \in [0, 1]$, $\sqrt{P^\lambda_{u,v}}$:

$$\sqrt{P^\lambda_{u,v}} = \prod_{i=1}^{N} \left(\frac{\sin((1-\lambda)d^i_G)}{\sin(d^i_G)} \sqrt{p^i_u} + \frac{\sin(\lambda d^i_G)}{\sin(d^i_G)} \sqrt{p^i_v} \right) \tag{5}$$

Here, $d^i_G = \cos^{-1}(\sum_{k=1}^{C} \sqrt{p^i_u(y_i = c|x_i)} \sqrt{p^i_v(y_i = c|x_i)})$. With the geodesic on this product manifold of *output* predictions in hand, the natural choice of a uniform *progress* coordinate, t_w, becomes immediate (t_w is the λ corresponding to the point along the geodesic that P_w is closest to):

$$t_w = \inf_{\lambda\in[0,1]} d_B(P_w, P^\lambda_{0,*}) \tag{6}$$

Here, d_B is the *Bhattacharya distance* between the distributions $P_w, P^\lambda_{0,*}$:

$$d_B(P_u, P_v) = -N^{-1} \sum_{i=1}^{N} \log \sum_{c=1}^{C} \sqrt{p_u^i(y_i = c|x_i)} \sqrt{p_v^i(y_i = c|x_i)} \tag{7}$$

Using the method of *Intensive Principle Components Analysis (InPCA)* [10], we may visualize these finetuning trajectories using a low-dimensional, isometric embedding designed precisely for such probabilistic models.

2.3. **Verifying Adversarial Robustness.** In addition to evaluating the transferability of various pretraining methods, we also consider the adversarial robustness of these methods. We ask whether contrastive learning produces representations that are more or less robust to adversarial attacks performed using the finetuned representation learned on the downstream task.

Fast Gradient Sign Method (FGSM): This is a simple attack method proposed in [2] that assumes knowledge of the model's weights to create adversarial examples. Adversarial examples are those that apply small perturbations to input images to create images that are misclassified by the classifier with high confidence.

The FGSM attack works as follows. Given an input datum x and associated label y, a loss function $\mathcal{L}(x, y; w)$ with w the model weights, we can create an adversarial sample $\tilde{x}$ by traversing along the directions that most increase the loss:

$$\tilde{x} \leftarrow x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(x, y; w), \epsilon \in (0, 1)) \tag{8}$$

In the case of an image, the resulting adversarial example is often visually very similar to the input image, but the model may be unable to classify these adversarial images correctly due to the particular nature of the perturbation.

3. Related Work

- [1] proposes the SimCLR framework using the InfoNCE loss. They find that using nearest neighbor classifiers on the encoder output leads SimCLR to perform similarly to supervised methods across a number of classification datasets. We choose to employ SimCLR as the benchmark for self-supervised contrastive learning.
- [6] proposes the SupCon loss that generalizes the methodology developed by SimCLR to scenarios in which labels are present, achieving similar levels of performance in comparison to supervised cross-entropy approaches. We choose to employ the SupCon loss as the benchmark for supervised contrastive learning.
- [11] develops the notion of training progress using a uniform time coordinate deriving from the geodesic of the product manifold of the model predictions that we utilize in this report.
- [5] verifies a number of transferability properties of contrastive learning methods on a variety of downstream tasks beyond classifications. The work finds that CL methods are less robust to adversarial perturbations, which we find surprising and seek to verify.

4. Approach

We structure the experiments in this report to provide empirical insight into the various properties discussed in the prevous sections. In particular, we look to compare the representations learned using 1) self-supervised CL (SimCLR), 2) semi-supervised CL (SupCon), and 3) supervised cross-entropy (SupCE) training methods.

We do so by *pretraining* ResNet-18 [3] backbones using CIFAR-10 [7] using each method and 4 random seeds. The pretrained models are then evaluated on 4 5-way classification tasks drawn from CIFAR-100 after training a linear classifier on top of the frozen backbone [1]. All experiments pretrain the network backbone for $T_p = 200$ epochs, and finetune for $T_f = 30$ epochs. All models are pretrained and finetuned using the SGD optimizer and a cosine annealing learning rate schedule. Experiment-specific details and results are provided in the following section. [2]

5. Experimental Results

We now highlight our key results, and expand on the key details in **Figures 1** and **2**. Unless otherwise stated, all results are on the validation datasets for the relevant downstream 5-way classification task on CIFAR-100.

- **Result 1**: Self-supervised (SimCLR) and supervised contrastive learning methods locate loss minima whose flatness is *comparable* to traditional cross-entropy based supervised learning on downstream cross-entropy based classification, despite using contrastive losses in the pretraining stage.

[1] Selected task labels: 1. $[0, 1, 2, 3, 4]$, 2. $[6, 11, 16, 21, 26]$, 3. $[56, 58, 62, 66, 68]$, 4. $[95, 96, 97, 98, 99]$

[2] Other relevant hyperparameters: $lr_{\max} = 0.05$ for SimCLR, SupCon, $lr_{\max} = 0.2$ for SupCE; $T = 20$ is the temperature parameter for SimCLR, SupCon, $\mathcal{B} = 256$, weight decay $\lambda = 10^{-4}$.

- **Result 2**: Self-supervised contrastive pretraining often makes more rapid progress towards the ground truth distribution, P^* during finetuning than supervised cross-entropy pretraining and even *ends* closer to the true distribution. *Supervised* contrastive pretraining appears to make slower progress than both self-supervised and cross-entropy methods.
- **Result 3**: We find that employing the SimCLR framework at the pretraining stage produced networks that were more robust to FGSM attacks compared to those pretrained using the supervised cross-entropy and supervised contrastive losses.

We briefly summarize the average validation accuracy across seeds for each of the 3 methods on each of the 5-way classification tasks at the end of the fine-tuning period in **Table 1**.

Training Method	CIFAR-100 [0, 1, 2, 3, 4]	CIFAR-100 [6, 11, 16, 21, 26]	CIFAR-100 [56, 58, 62, 66, 68]	CIFAR-100 [95, 96, 97, 98, 99]
SimCLR	79.95	**86.11**	92.55	**90.91**
SupCon	78.20	78.85	89.53	88.26
SupCE	**82.48**	82.50	**93.50**	88.79

TABLE 1. Validation accuracies of SimCLR, SupCon, and SupCE after finetuning on 5-way classification tasks drawn from CIFAR-100 (labels indicated) averaged across 4 training seeds. Observe that SimCLR occasionally outperforms traditional supervised cross-entropy on tasks, while all methods achieve fairly high task accuracies, implying the viability of contrastive pretraining.

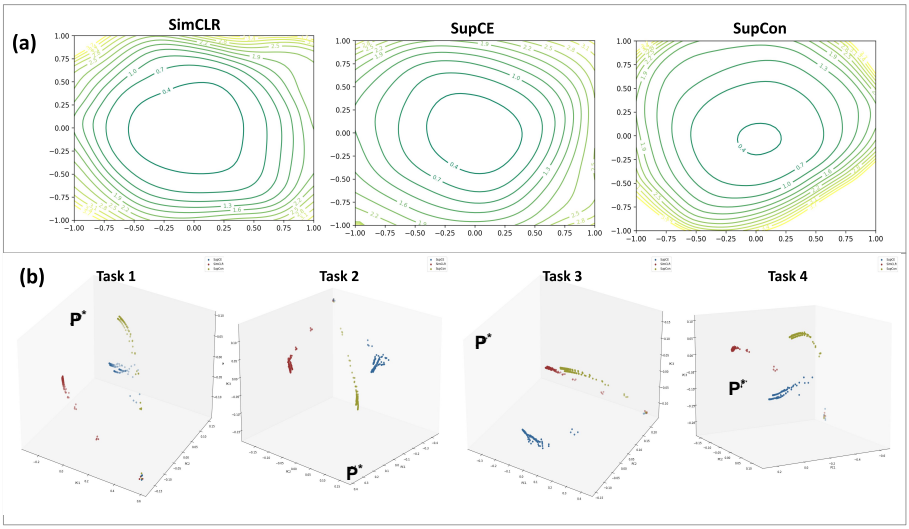

FIGURE 1. **(a) Loss landscape visualization at end of 5-way classification fine-tuning for SimCLR, SupCon, and SupCE on Task 4 – [95, 96, 97, 98, 99]**. Using the method described in [8], we are able to visualize the loss landscape along two randomly chosen directions in weight space using the "filter-wise normalization" technique. We note that all 3 pretraining approaches induce similarly flat loss landscape, which, by the arguments made in [9], implies they are *similarly* transferrable despite using entirely different pretraining strategies. It is particularly noteworthy considering contrastive losses are not the cross-entropy based losses used during finetuning, nor do these losses necessarily require labeled data.
(b) InPCA training trajectories for SimCLR, SupCon, and SupCE for all 4 5-way classification tasks, trajectories for all seeds shown (Red = SimCLR, Yellow = SupCon, Blue = SupCE). By using InPCA on the probability vectors of network predictions at each epoch during the fine , we are able to visualize the trajectory taken by the model from the initialization of the linear classifier to the end of finetuning, as well as the location of P^*, the ground truth distribution. All models seem to take entirely different trajectories. This is a point that is not simply due to random initialization – random seeds corresponding to the same loss follow nearly identical trajectories. Still, all approaches seem to make monotonic progress towards P^*, a point we expand on in Fig. 2.

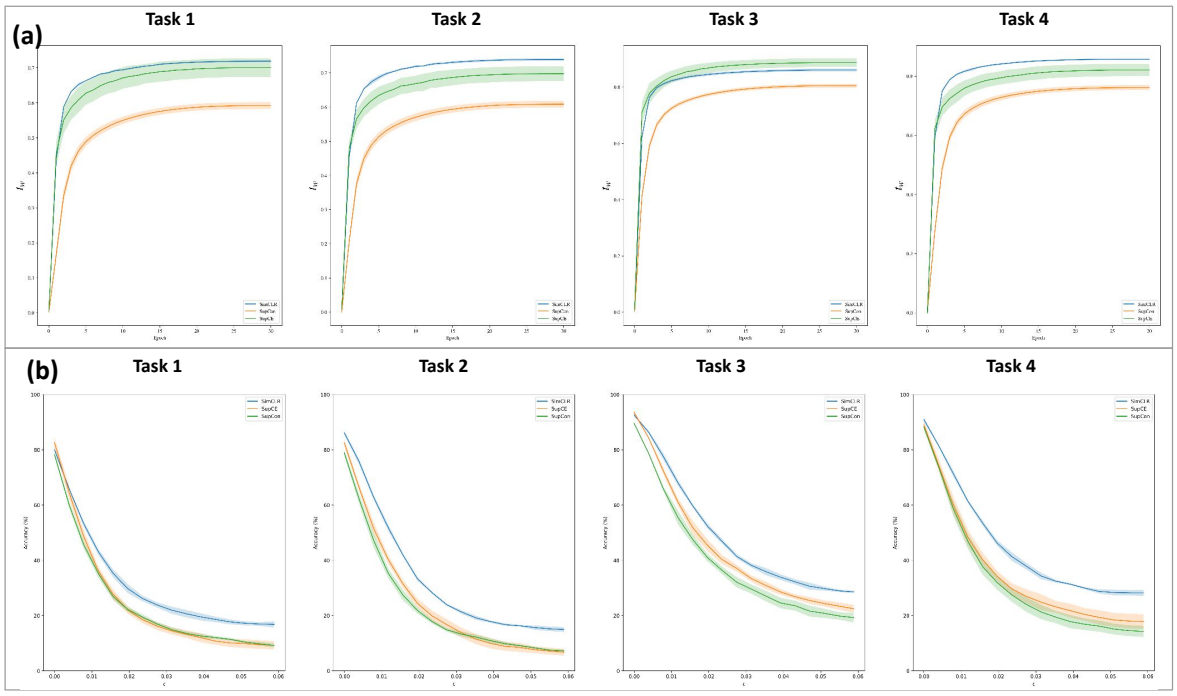

FIGURE 2. **(a) Progress (t_w) along geodesic of product manifold for SimCLR (Blue), SupCon (Orange), and SupCE (Green) over finetuning period on CIFAR-100 5-way classifications**. We note that using the measure of progress determined by the uniform time coordinate t_w given by Eqn. 6, SimCLR outperforms traditional cross-entropy based pretraining on 3 of 4 tasks even when accounting for the marginal uncertainty induced by random seeds. This further demonstrates the strong transferability of self-supervised contrastive learning based models on a number of downstream tasks. One suspects that the information induced by the labels in SupCE actually hinders transferability since the pretraining labels have little connection to the finetuning labels.
(b) Effect of FGSM attacks on accuracy with CIFAR-100 5-way classification tasks, ϵ varies from $1/255$ to $15/255$ (Blue = SimCLR, Orange = SupCE, Green = SupCon). Contrary to expectations suggested by [5], we note that SimCLR appears more robust to FGSM attacks than both supervised cross-entropy and the supervised contrastive loss. However, we believe this makes sense – the improved transferability of the model from task to task *should* contribute to improved robustness to perturbations in the input space. This is because the model is well-conditioned for a number of tasks, and thus it is similarly well-conditioned for a number of inputs.

6. CONCLUSION

In this work, we have extensively verified both the transferability and robustness properties of contrastive learning methods on various 5-way classification tasks, and perform extensive benchmarking to more traditional cross-entropy based methods. We show that pretraining using contrastive learning, particularly self-supervised approaches, like the SimCLR framework can frequently *outperform* cross-entropy based methods.

This suggests that contrastive learning is a viable, and perhaps even preferrable method for pretraining large networks, particularly in the context of visual representation learning. In addition, these approaches require little to no labeled data, making them more readily deployable. Implementing and optimizing such models is also comparatively straightforward (say in comparison to *generative* models like GANs).

Future work would consider both exploring the performance of contrastive methods on other kinds of image-based tasks (segmentation, object detection, etc.). In addition, we would like to explore the geometry of the weights themselves more carefully by examining properties of the network's Hessian or Fisher Information matrix.

REFERENCES

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[2] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[4] Weiran Huang, Mingyang Yi, and Xuyang Zhao. Towards the generalization of contrastive self-supervised learning, 2021.

[5] Ashraful Islam, Chun-Fu Richard Chen, Rameswar Panda, Leonid Karlinsky, Richard Radke, and Rogerio Feris. A broad study on the transferability of visual representations with contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8845–8855, 2021.

[6] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc., 2020.

[7] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[8] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Neural Information Processing Systems*, 2018.

[9] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael I. Jordan. Towards understanding the transferability of deep representations, 2019.

[10] Katherine N. Quinn, Colin B. Clement, Francesco De Bernardis, Michael D. Niemack, and James P. Sethna. Visualizing probabilistic models and data with intensive principal component analysis. *Proceedings of the National Academy of Sciences*, 116(28):13762–13767, 2019.

[11] Rahul Ramesh, Jialin Mao, Itay Griniasty, Rubing Yang, Han Kheng Teoh, Mark Transtrum, James P. Sethna, and Pratik Chaudhari. A picture of the space of typical learnable tasks, 2022.