Adversarial Augmentations for Contrastive Learning

Ekdeep Singh Lubana
University of Michigan
eslubana@umich.edu

Bahaa Aldeeb
University of Michigan
baldeeb@umich.edu

Abstract

The success of contrastive learning is contingent on generating strong views of a sample that help promote invariances to spurious features (e.g., blue backgrounds in fish images). This necessitates the design of non-trivial augmentation policies, a component that is still heuristically configured in contrastive learning frameworks. To address this, we propose to "learn to augment" a sample. Specifically, we use adversarial training *to generate samples that maximally increase the contrastive loss. The hypothesis we want to test is whether learning to be invariant to such strong views, which maximally increase the loss, will also result in invariance to relatively simpler views, correspondingly improving downstream performance. We share both negative and positive results on this hypothesis, while carefully designing a framework that makes learned augmentations possible for contrastive learning.*

1. Introduction

Contrastive learning, a self-supervised learning methodology which uses instance discrimination as a pretext task for representation learning, has been highly successful at achieving performances similar to supervised learning [2, 7, 12]. In general, contrastive learning frameworks seek to promote similarity between features extracted from two *positively* related samples (called "positive views"; e.g., sample from same class), while penalizing similarity between features extracted from *negatively* related samples (called "negative views"; e.g., samples from different classes). However, without the knowledge of underlying label information, the positive/negative relationship between two samples cannot be ascertained easily.

To circumvent this problem, most contrastive learning frameworks use data augmentation strategies to convert an input into a positively-related sample, while considering all other samples in the dataset to be negatively related [2]. Some recent frameworks even demonstrate the capability to forego negative samples altogether [6, 3], achieving high performance on downstream tasks while relying on positive samples only. Thus, one can unarguably see that the common thread underlying the success of effective contrastive learning frameworks is *strong data augmentations*. Strong data augmentations enable a model to learn to be invariant to spurious features that are unlikely to be useful for downstream tasks. For example, using color jittering as an augmentation can allow a model to avoid using the constant, blue background in fish images as a classification feature, hence learning useful structural information about the object of interest. This information can then be used for downstream tasks, such as image classification.

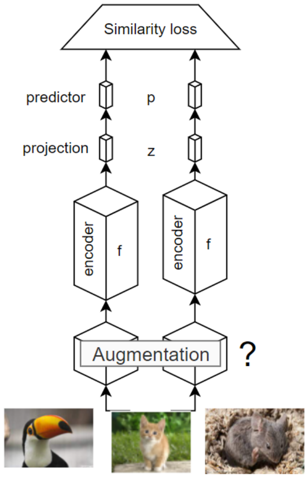

Figure 1. We explore the idea of "learning" how to augment in this report. Our hypothesis is that using a data driven approach to automatically augment is more likely to find an optimal augmentation compared to hand-crafted methods. The main framework we use to this end is adversarial training, using which "difficult" samples can be learned. By performing well on these difficult samples, we expect to improve performance on "easier" samples and hence on the overall downstream task (classification in this report) too.

In this work, we undertake a study of "learning to augment" for contrastive learning. Our hypothesis is that given the immense value of data augmentation, hand-crafted augmentation strategies are unlikely to be the optimal choice for generating positive views of a sample. To probe this hypothesis, we propose to use adversarial training to automatically generate input views that maximally increase the contrastive loss. Our intuition is as follows: Since contrastive losses focus on maximizing representational similarity between two views $\{X_1, X_2\}$ of an input X, adversarial training will amount to designing a view (say, X_1) that has a highly dissimilar representation w.r.t. the other view (i.e., X_2). Training with such an adversarial pair will thus force the model to learn to be invariant to strong input perturbations, hopefully leading to better performance on less perturbed inputs and hence on downstream tasks as well.

We note that the idea of adversarial training for data augmentation in contrastive learning has indeed been explored in the past [9, 11, 8]. However, in this report, we demonstrate that the naive use of adversarial training for data augmentation suffers from several flaws: (i) For frameworks which use negative samples, adversarial training can be shown to generate views that maximize representational similarity with false negatives (i.e., negative samples sharing the same label). This has provable adverse effects on downstream performance, as shown by Arora et al. [1]. (ii) Adversarial perturbations are unstructured by default and the resulting samples are often very similar to the original examples, providing very minimal training signal for contrastive learning. We address these limitations by (i) focusing on a positives-only framework (specifically, SimSiam) and (ii) using structured adversarial perturbations (e.g., learning colorspace transformations).

2. Related Work

Learning to Augment: Learning to augment has been extensively explored in the context of supervised learning [4, 5, 22]. For example, Cubuk et al. [4] propose to use an RNN-based controller to learn an augmentation policy that helps achieve the best performance on proxy tasks that are closely aligned with the target tasks. We differ from this body of work by focusing on unsupervised learning, where labels for determining the "optimal" policy are not available, resulting in a non-trivial problem.

Hard-Negative Mining: Finding hard negatives for representation learning has a rich history in the field of metric learning [21]. Due to the similarities between contrastive learning and metric learning, hard negative mining has been employed in contrastive learning as well [10, 17, 19]. However, we stress our objective in this work is different from negative mining: we intend to find *strong positives*, i.e., good augmentations of a sample.

Adversarial Data Augmentation: Adversarial training has seen popular use as a data augmentation strategy in semi-supervised learning [15], where representations of a model trained on minimal labeled data are used as pseudo-labels for training another model on unlabeled data. In this vein, recent works have also proposed to employ adversarial data augmentation in negative-sample contrastive learning frameworks [9, 11, 8]. However, as we show in the following, a naive use of adversarial training suffers from two drawbacks: (a) in negative-sample frameworks, it increases the adverse effects of false negatives and (b) due to the unstructured nature (see section 4) of adversarial perturbations, its is unable to improve downstream performance.

3. Pitfalls in Adversarial Augmentation

We begin by discussing pitfalls in naive use of adversarial augmentation for contrastive learning. To this end, we first establish necessary notations for adversarial training, following Madry et al. [14]:

Notations: Consider a model $f(\theta)$ trained on a dataset $\mathcal{D}$ using a loss function $L(f(\theta); \mathcal{D})$. Then, under adversarial training, one first samples an input X from the data and learns a perturbation δ such that adding this perturbation to the input maximally increases the model loss. The model is trained to minimize loss on both clean samples X and the perturbed samples $X + \delta$. These perturbed samples thus serve the function of data augmentation. Generally, one constrains δ to lie within a ball of ϵ radius, where distance is usually measured using $||.||_\infty$. Formally, we have:

$$\begin{aligned} \delta^* &= \arg \max_{\delta:||\delta||_\infty < \epsilon} L(f(\theta); X + \delta) \quad \text{(max-step)}, \\ \theta^* &= \arg\min_\theta L(f(\theta); X + \delta^*) \quad \text{(min-step)}. \end{aligned} \tag{1}$$

3.1. Adversarial Training with Negative-Sample Frameworks

We use SimCLR [2] as a representative example of negative-sample frameworks. Recall that SimCLR samples an input X and a positively-related sample X^+ (both are random augmentations of a raw sample) and processes them using two models–first a backbone model and then a projector model. The projector outputs normalized representations, $\{z, z^+\}$. Then, representations from $K-1$ negative samples $z^-_{i \in 1 \dots K-1}$ are used to compute the following variant of InfoNCE loss [18]:

$$L(z) = -\log\left(\frac{e^{\text{sim}(z,z^+)}}{e^{\text{sim}(z,z^+)} + \sum_{i=1}^{K-1} e^{\text{sim}(z,z_i^-)}}\right), \tag{2}$$

where the functional sim(., .) is generally the euclidean inner product. Note that the loss is made symmetric by also adding a component corresponding to z^+, but we focus on the above described asymmetric version for now.

Since our goal is to learn to augment, we aim to adversarially learn a perturbation δ, which upon addition to sample X will maximally increase model loss. To understand what this means, we compute the gradient of the above loss with respect to representation z:

$$\nabla_z L = -(1 - P^+) z^+ + \sum_{i=1}^{K-1} P_i^- z_i^-, \tag{3}$$

where $P^+ = e^{\text{sim}(z,z^+)}/(e^{\text{sim}(z,z^+)} + \sum_{i=1}^{K-1} e^{\text{sim}(z,z_i^-)})$ is the probability that z is more similar to z^+ than to other samples. Similarly, $P_i^- = e^{\text{sim}(z,z^-)}/(e^{\text{sim}(z,z^+)} + \sum_{i=1}^{K-1} e^{\text{sim}(z,z_i^-)})$ represents the probability that z is more similar to z_i^- than to other samples.

Since adversarial training computes perturbation δ to maximize the loss, the above relationship shows that the computed perturbation will achieve its objective by reducing the component of representation in z that overlaps with z^+, while increasing the component in the direction of z^- that is closest to z. In general, since samples are retrieved randomly from a dataset, there exist "false negatives" in the set of negatives $z^-_{\{1,\ldots,K-1\}}$. Specifically, if the input belongs to a class C, it is very likely that a randomly sampled batch of samples will have other samples from class C too. Since all samples in a batch serve as negatives to each other, this implies we will end up considering two samples of the same class to be negatives of each other, which is not the case. Such instances are called False Negatives.

As one may expect, false negatives of z have the most similar representations with respect to z throughout training [10]. Arora et al. [1] show the presence of such samples provably hurts representation quality for downstream tasks. Thus, by using adversarial training in negative-sample frameworks and forcing the learned sample to become more similar to its closest false negatives, one is implicitly encouraging the contrastive signal between the two to grow stronger. This will exacerbate the adverse effects of false negatives and is likely to hurt downstream performance. Indeed, we see that earlier work on adversarial training for negative-sample frameworks shows significant performance loss [9]. Specifically, on a simple dataset of CIFAR-10, a loss of almost 3% is observed.

Main Takeaway: Our analysis demonstrates the use of adversarial training in negative-sample contrastive learning will hurt performance on downstream tasks, as has been observed in prior work [9, 11]. Given the objective of learned data augmentations is to improve performance, preventing performance loss is certainly not sufficient.

3.2. Adversarial Training with Positive-Sample Frameworks

In frameworks that rely on positive samples only, one cannot run into the problem of false negatives. This makes positive-sample frameworks a viable target for adversarial data augmentation. In the following, we use SimSiam [3] as a representative example of positive-sample frameworks.

Recall, SimSiam samples an input X, transforms it to determine two augmentations $\{X_1, X_2\}$, and uses three models to extract representations: (i) a backbone model, which will be used for downstream tasks; (ii) a projector model, whose representations are denoted as $\{z_1, z_2\}$; and (iii) a predictor model, whose representations are denoted as $\{p_1, p_2\}$. The SimSiam loss is then computed as follows:

$$L(X_1, X_2) = -\text{sim}(p_1, \text{sg}(z_2)) - \text{sim}(p_2, \text{sg}(z_1)), \quad (4)$$

where sg(.) denotes a stop-grad operation and sim(., .) denotes the cosine similarity between two vectors.

Now, say the view X_1 is to be adversarially augmented, then computing the gradient of the above loss with respect to z_1, we see the following:

$$\nabla_{z_1} L^T = -\nabla_{z_1} \text{sim}\left(p_1, \text{sg}(z_2)\right). \quad (5)$$

That is, due to the presence of the stop-grad operation, the second half of the loss plays no role in providing a signal for adversarial augmentation. We thus use the above loss for finding adversarial perturbations for a given sample X_1, while keeping its corresponding positive view X_2 fixed.

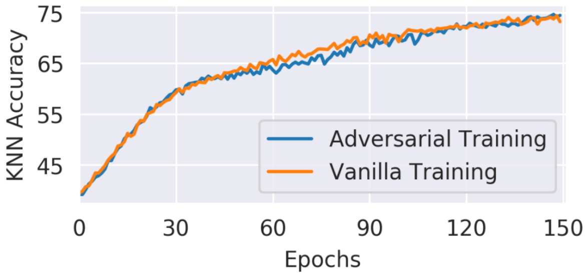

Figure 2. KNN accuracy of SimSiam trained using adversarial augmentations. We find positive-sample frameworks do not suffer performance loss under adversarial training. This is in contrast with negative-sample framework, which suffer performance loss [9] because of exacerbated adverse effects of negative samples via adversarial training (see subsection 3.1)

Observation: Adversarial training on SimSiam matches vanilla training's performance. We use a ResNet-12 with half the number of filters per layer to train a SimSiam model on CIFAR-10 for 150 epochs. The learned representations are used to evaluate KNN accuracy on test data to track representational quality during training [20], with K=200, as used in SimSiam. Results are shown in Figure 2. As can be seen, the model is able to match the performance of vanilla training (with moments of minimal performance loss), unlike negative-sample frameworks, providing corroboration to our claim that increased adverse effects of false negatives due to adversarial training hurts representational quality. Nonetheless, we still do not see an improvement in performance, our desired objective!

Main Takeaway: Positive-sample frameworks are not susceptible to adverse effects of false negatives, unlike negative-sample frameworks. However, adversarial training does not result in any performance gains either.

4. Structured Adversarial Training

While the results on positive-sample frameworks seem counter-intuitive at first, probing into past adversarial train-

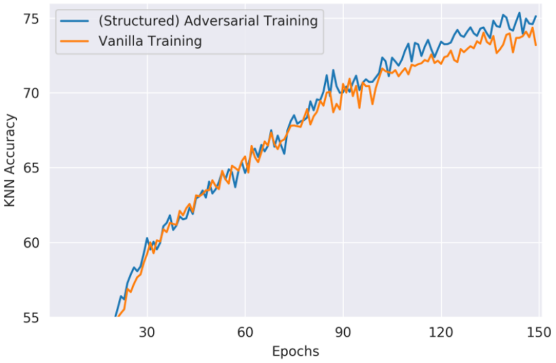

Figure 3. KNN accuracy of SimSiam trained using structured adversarial augmentations (specifically, interpolating colorspace augmentation). We see a non-trivial, consistent performance boost, hence providing validation for our hypothesis.

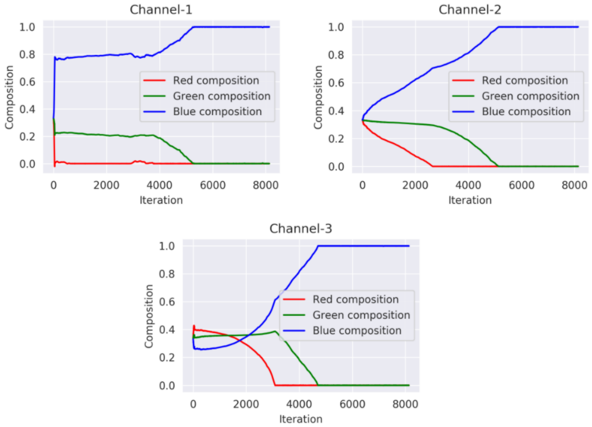

Figure 4. Progression of interpolation constants for interpolating color augmentation over training. Interestingly, we find the model chooses to heavily rely on blue-channel information.

ing literature, we find that adversarial training is known to decrease the performance of a model on clean samples. This partly ensues because invariance to adversarial noise seeks to increase the model's performance in regions of close proximity, even if those regions may contain inputs that are impossible to exist naturally. This is a repercussion of adversarial gradients being unstructured in nature under the loose constraint of $||\delta||_\infty < \epsilon$.

To address this, we take inspiration from the work of [13], who propose to perform adversarial training in a structured manner. In brief, the idea is to project the adversarial gradient into structured spaces which are plausible to occur naturally. For example, consider rotation of a sample. One can "learn" the degree of rotation in an adversarial way, such that the rotated sample is difficult for the model to perform well on. Such samples remain close to being natural and performing well on them is likely to help generate robust, high-quality representations for downstream tasks.

4.1. Experiments

To probe the proposed idea of structured adversarial training, we first consider a simple augmentation strategy: Interpolating color transformations. Specifically, we learn an interpolating color space that takes the RGB information at a pixel and outputs a 3-tuple (C_1, C_2, C_3) that corresponds to a "learned" color space. Formally, we have:

$$C_i = \alpha_{i,1}R + \alpha_{i,2}G + \alpha_{i,3}B, \tag{6}$$

such that $0 \leq \alpha_{i,j} \leq 1$ and $\sum_j \alpha_{i,j} = 1$. Note that one can simply differentiate with respect to the above scalars using autodiff and correspondingly update them in an adversarial manner. We use a 1-layer MLP to parameterize our adversarial model. The models use backbone representations as input and use a Softmax to satisfy the constraint on transformation constants $\alpha_{i,j}$.

We use the same experimental benchmark as in figure Figure 2. Results are shown in figure Figure 3. As can be seen, KNN accuracy shows a consistent, non-trivial improvement trend with respect to the vanilla SimSiam training. *These results provide validation for our claim that adversarial training, when used in a structured manner, can "learn" to generate augmentations that provide useful performance gains for contrastive learning.*

What are we learning? To probe the representations that the model has learned, we track the weights for our interpolating colorspace transformation (see Figure 4). Interesetingly, we find the model learns to output completely blue images. Blue is a rare color in nature, outside of sky and sea. This might indicate that when this augmentation has reached an extreme it starts distracting from colors that carry most useful features. Naturally, if that is the case, blue images will yield higher loss. This suggests that more constrained augmentation might even be of greater benefit.

5. Conclusion and Future Work

Contrastive learning benefits more from structured augmentations that emphasize potentially natural yet spurious features in a positive pair. It is evident that better informed forms of such augmentations can be beneficial for positive-sample based contrastive learning. In this work simple learning methods were use. Further ablations on the types and extent of augmentation might yield better results.

We also note that using a recently released differentiable data augmentation library [16], we were able to perform other interesting augmentations as well (e.g., Rotation, color jittering, Translation). The results look very similar to the colorspace transform results and in the interest of space, we could not include them. Further, the library supports 3D data augmentations and it may be interesting to try our hypothesis on 3D data, where data is available in only limited amounts generally.

References

[1] S. Arora, H. Khandeparkar, M. Khodak, O. Plevrakis, and N. Saunshi. A Theoretical Analysis of Contrastive Unsupervised Representation Learning. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2020. 2, 3

[2] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2020. 1, 2

[3] X. Chen and K. He. Exploring Simple Siamese Representation Learning. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 3

[4] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. AutoAugment: Learning Augmentation Policies from Data. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[5] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. RandAugment: Practical automated data augmentation with a reduced search space. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[6] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, and M. Valko. Bootstrap Your Own Latent-A New Approach to Self-Supervised Learning. In *Proc. Adv. in Neural Information Processing Systems (NeurIPS)*, 2020. 1

[7] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[8] C.-H. Ho and N. Vasconcelos. Contrastive Learning with Adversarial Examples. In *Proc. Adv. in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[9] Z. Jiang, T. Chen, T. Chen, and Z. Wang. Robust Pre-Training by Adversarial Contrastive Learning. In *Proc. Adv. in Neural Information Processing Systems (NeurIPS)*, 2020. 2, 3

[10] Y. Kalantidis, M. B. Sariyildiz, N. Pion, P. Weinzaepfel, and D. Larlus. Hard Negative Mixing for Contrastive Learning. In *Proc. Adv. in Neural Information Processing Systems (NeurIPS)*, 2020. 2, 3

[11] M. Kim, J. Tack, and S. J. Hwang. Adversarial Self-Supervised Contrastive Learning. In *Proc. Adv. in Neural Information Processing Systems (NeurIPS)*, 2020. 2, 3

[12] J. Li, P. Zhou, C. Xiong, and S. C. Hoi. Prototypical Contrastive Learning of Unsupervised Representations. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2021. 1

[13] C. Luo, H. Mobahi, and S. Bengio. Data Augmentation via Structured Adversarial Perturbations, 2020. 4

[14] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2018. 2

[15] T. Miyato, S. ichi Maeda, M. Koyama, and S. Ishii. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018. 2

[16] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski. Kornia: an Open Source Differentiable Computer Vision Library for PyTorch. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

[17] J. D. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka. Contrastive Learning with Hard Negative Samples. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2021. 2

[18] A. van den Oord, Y. Li, and O. Vinyals. Representation Learning with Contrastive Predictive Coding. In *Proc. Adv. in Neural Information Processing Systems (NeurIPS)*, 2018. 2

[19] M. Wu, M. Mosse, C. Zhuang, D. Yamins, and N. Goodman. Conditional Negative Sampling for Contrastive Learning of Visual Representations. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2021. 2

[20] Z. Wu, Y. Xiong, S. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance-level discrimination. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[21] H. Xuan, A. Stylianou, X. Liu, and R. Pless. Hard negative examples are hard, but useful. In *Proc. Euro. Conf. on Computer Vision (ECCV)*, 2020. 2

[22] X. Zhang, Q. Wang, J. Zhang, and Z. Zhong. Adversarial AutoAugment. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2020. 2