Improve Model Robustness in Less Time Than It Takes to Drink A Cup of Coffee with Plug-and-Play Robustness Plugins

Jiang Fang[1,2], Zhicheng Zhang[1,2], Jiyan Sun[1,*], Jiadong Fu[1,2], Haonan He[1,2], Yinlong Liu[1,2], and Wei Ma[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyberspace Security, University of Chinese Academy of Sciences, Beijing, China
*Corresponding Author: sunjiyan@iie.ac.cn

Abstract. Self-supervised learning has become the primary method for pre-training large models due to its ability to train without labeled data and its excellent data feature representation capabilities. However, neural network models are vulnerable to adversarial attacks, which can lead to incorrect predictions. Previous work has attempted to enhance the robust representation capabilities of base models through self-supervised adversarial training (self-AT), which integrates adversarial training into the self-supervised learning pre-training process. However, self-supervised learning requires numerous training epoches, and adversarial training is computationally complex. Consequently, these methods need an additional 2.75 to 12 times the pre-training time of the model to obtain robust representations. Considering the resource consumption of training large models and the current high cost of computational resources, the cost of obtaining robustness for base models is excessively high and impractical. This paper proposes a novel Plug-and-Play model Robustness Plugin training framework called PPRP. PPRP is designed as a robustness plugin for self-supervised base models that have completed pre-training. Once the robust plugin is added, the base model gains robust representation capabilities. Essentially, PPRP is a teacher-student network that performs adversarial training on a plugin model with only a few parameters, reducing the time required to achieve model robustness to 5% of the pre-training time. The robust plugin can be seamlessly integrated into pre-trained models without additional inference latency. Experiments show that on multiple datasets, different base models with the PPRP-trained robust plugin achieve state-of-the-art robustness.

Keywords: Adversarial Training · Self Supervised Learning · Robustness Plugin · Low-Rank Adaptation

1 Introduction

Self-Supervised learning (SSL) [1, 3, 4, 10, 12] can leverage a large amount of high-quality unlabeled data to train base models with good multi-task transfer

capabilities. As the amount of data involved in training increases, the size of the self-supervised models also increases exponentially to learn the intrinsic representations of the massive data, such as GPT series [2, 19, 20]. Similar to supervised learning, neural network models trained through self-supervised learning are also susceptible to adversarial attacks [9]. Adversarial training (AT) [18] is the de facto standard method for achieving adversarial robustness in models.

Several works [8, 14, 15, 17] have successfully integrated adversarial training into the pre-training process of SSL to enhance the adversarial robustness of pre-trained base models. However an adversarial training process contains multiple forward and backward propagations to compute the adversarial samples, and SSL itself requires long training epochs (often up to thousands of epochs). Integrating adversarial training into pre-training process greatly extends the training time and the consumption of training resources.

If we define the ratio of the difference between the total time required for self-supervised adversarial training (R_t) and the time required for self-supervised training alone (S_t) to the time required for self-supervised training alone as the Robustness Acquisition Time Cost Ratio (RATCR) of the model ($(R_t - S_t)/S_t$). Previous works [8,14,15,17] have shown RATCRs ranging from 2.75 to 12, meaning that self-supervised adversarial training (self-AT) requires an additional 2.75 to 12 times the training time to achieve model robustness. Considering the current trend of increasing model sizes and the high cost of computing resources, this method of acquiring robustness is unacceptable. Particularly considering that edge networks such as power distribution grids widely use resource-constrained edge IoT devices, it is even more unrealistic to train robust models on these edge devices to ensure data flow security using the aforementioned methods.

Recently, DeACL [22] proposed enhancing the robustness of base models through adversarial training after the completion of base model pre-training. Although DeACL effectively reduces the time required to enhance the robustness of the base model, lowering the RATCR to 0.7, it still requires updating all the parameters of the base model and does not alleviate the computational resource consumption associated with enhancing model robustness.

To address above challenges, this paper proposes a novel **P**lug-and-**P**lay model **R**obustness **P**lugin training framework for trained self-supervised learning base models, named PPRP. Structurally, PPRP is a teacher-student network, where the pre-trained base model serves as the teacher model, and PPRP conducts adversarial training on a robustness plugin with only a small number of parameters. The robustness plugin with a small number of parameters acts as the student model, receiving adversarial samples as input, while the teacher model receives the original samples. PPRP achieves adversarial robustness by aligning the outputs of the student model with those of the teacher model. PPRP utilizes LoRA [13], a parameter-efficient fine-tuning (PEFT) method, to construct the robustness plugin model, with the plugin's parameters accounting for only 1% of the base model's parameters. These robustness plugin parameters can be directly merged with the base model parameters, without increasing the inference time of the base model. For real-time operational systems like power distribu-

tion grids, PPRP can be seamlessly integrated into pre-trained models without adding inference latency. This means that it can defend against adversarial attacks while ensuring the security of the data in power distribution grids without compromising the efficiency of real-time data transmission.

PPRP can easily train robustness plugins for different self-supervised learning base models. After a few training epochs, the base model with the added robustness plugin achieves state-of-the-art robustness. PPRP reduces the Robustness Acquisition Time Cost Ratio (RATCR) from 0.68 to 0.046, a dramatic 15-fold decrease. PPRP reduces the time required to enhance the robustness of base models from several hours or even days to just 22 minutes. Now, you can add robustness to your base model in the time it takes to enjoy a cup of coffee. Overall, the contributions of this paper are as follows:

- We propose a novel Plug-and-Play Robustness Plugin training framework for self-supervised learning base models. The proposed framework enables PPRP to train a small robustness plugin, with plugin parameters accounting for only 1% of the base model's parameters. Adding this plugin to the base model can significantly improve the model's robustness.
- PPRP can seamlessly adapt to various SSL base models and is highly efficient in training, reducing the Robustness Acquisition Time Cost Ratio (RATCR) from 0.68 to 0.048, a dramatic 15-fold decrease.
- Experimental results demonstrate that base models with the robustness plugin trained by PPRP achieve state-of-the-art robustness on multiple datasets, thereby substantiating the effectiveness and superiority of PPRP.

2 Related Work

In this section, we will first introduce related work on self-supervised learning, followed by recent advancements in the use of adversarial training to endow self-supervised learning models with adversarial robustness.

2.1 Development in SSL

From natural language processing [7,20] to computer vision [3,12], self-supervised learning has undoubtedly become the preferred method for pre-training large models due to its ability to learn from data without Ground Truth labels. Contrastive learning [1,3,4,10–12] is a popular approach in self-supervised learning for visual model pre-training. Contrastive learning learns data representations by ensuring the consistency of different augmented versions of the same sample (positive samples). For example, the objective of SimCLR [3] is to bring positive samples closer together in the representation space while pushing apart views of different samples (negative samples). To avoid the computational resource consumption caused by using negative samples in SimCLR, some methods attempt to complete model pre-training using only positive samples. For instance, BYOL [10] and SimSiam [4] use asymmetric network structures, directly predicting the representation of one positive sample from another. Barlow Twins [21]

and VICReg [1] aim to bring positive sample representations closer while ensuring the representations of different samples maintain diversity.

2.2 Development in self-AT

Adversarial training (AT) [18] is a game between an attacker and a defender. The attacker calculates adversarial examples that maximize the training loss, while the defender incorporates these adversarial examples into the training set to optimize the model and minimize the training loss. Previous works have focused on supervised adversarial training, but recent studies have begun to introduce adversarial training into self-supervised learning. RoCL [15], ACL [14], AdvCL [8], and DynACL [17] integrate adversarial training into the pre-training process of SimCLR. They first compute adversarial examples that maximize the SimCLR loss and then minimize the model's loss on these adversarial examples. Since self-supervised learning requires multiple training epochs and the computation of adversarial examples involves several iterations, integrating adversarial training into the pre-training process significantly prolongs the training time for robust self-supervised base models. The most similar work to ours is DeACL [22], which does not integrate adversarial training with the pre-training process. Instead, DeACL divides SSL and AT into two stages. DeACL uses the output of the pre-trained model as a pseudo-target, and uses the pseudo-target with the adversarial training to train a new base model, which has a much higher number of training parameters than the PPRP due to the fact that DeACL requires the update of all the parameters of the base model.

3 Method

During the PPRP training process, only the parameters of the robustness plugin are updated. By adopting the parameter efficient fine tuning method, LoRA [13], PPRP imparts excellent robustness to the model by updating only a small number of parameters. Essentially, PPRP is a teacher-student network, where the output of the pre-trained model serves as the teacher model's output, guiding the learning of the robustness plugin.

3.1 PPRP framework

Given a trained self-supervised pre-training base model f_θ, where θ represents the parameters of the pre-trained base model, and a robustness plugin model $f_{\Delta\theta}$, where $\Delta\theta$ represents the parameters of the robustness plugin, the training objective of PPRP is as follows

$$\begin{aligned}\arg\min_{\Delta\theta}[&\ell_{mse}(f_{\theta+\Delta\theta}(x), f_\theta(x))+\\ &w\cdot\ell_{mse}(f_{\theta+\Delta\theta}(x+\delta), f_\theta(x))]\end{aligned} \tag{1}$$

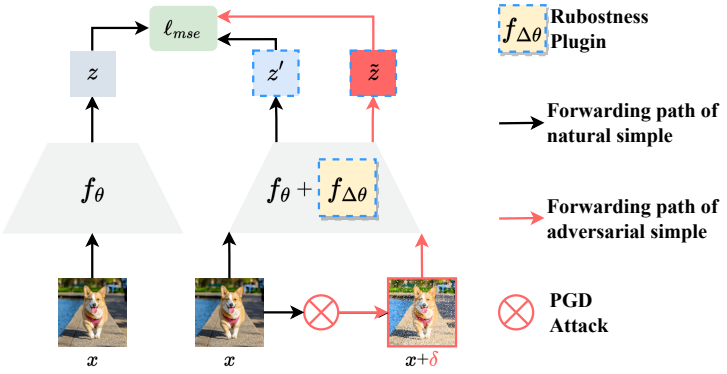

Fig. 1: Plug-and-Play Robustness Plugin training framework. f_θ is the trained base model and $f_{\Delta\theta}$ is the robustness plugin. δ is an adversarial perturbation under PGD attack. The base model parameters are frozen during training and only the robustness plugin parameters are updated.

Here, x is the original input sample, and $x + \delta$ is the adversarial sample obtained by adding adversarial perturbation δ to sample x, ℓ_{mse} represents the Mean Squared Error (MSE).

PPRP generates adversarial perturbations as follows

$$\delta := \arg \max_{\delta;\|\delta\| \leq \eta} \ell_{mse}(f_\theta(x), f_{\theta+\Delta\theta}(x+\delta)), \tag{2}$$

where η denotes the allowed range of adversarial perturbations. According to DeACL [22], the adversarial perturbation δ is computed using a 5-step PGD (with a step size of $\epsilon = 8/255$). Clearly, Equation 1 consists of two terms: the first term ensures that the pre-trained base model maintains excellent representation performance after adding the robustness plugin, while the second term ensures that the robustness plugin effectively enhances the robustness of the original pre-trained base model. w is a hyperparameter used to balance accuracy and robustness.

3.2 Plugin parameters update

To reduce the training parameters of the robustness plugin and thereby decrease the consumption of expensive computational resources during the training process, PPRP employs LoRA, a parameter-efficient fine-tuning method, to update the parameters of the robustness plugin. According to LoRA, we decompose the plugin model parameters $\Delta\theta$ into the product of two low-rank matrice A and B.

$$\theta + \Delta\theta = \theta + BA \tag{3}$$

where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times d}$. Since $r \ll d$, the number of parameters in the plugin, $d \times r + r \times d$, is much smaller than the number of parameters in the

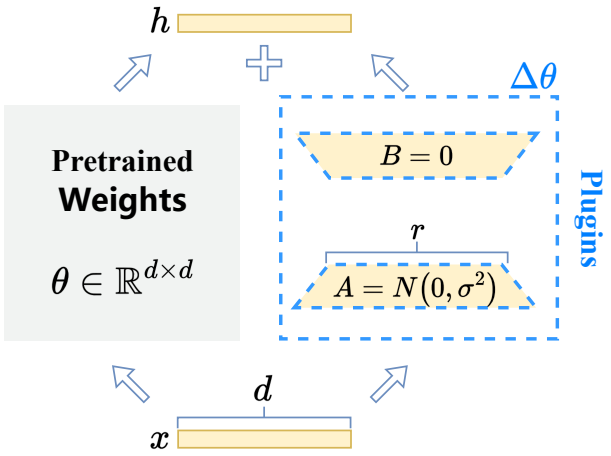

Fig. 2: PPRP Plugin Parameter Update Mechanism.

original model, θ, which is $d \times d$. Given the input x, the output of the base model with the plugin added is

$$h = \theta \cdot x + \Delta\theta \cdot x = (\theta + \Delta\theta) \cdot x \tag{4}$$

The base model parameters θ are frozen during training and only the plugin parameters $\Delta\theta$ are updated.

4 Experiments

In this section, to validate the effectiveness of PPRP, we will first compare its robustness against baseline methods RoCL [15], ACL [14], AdvCL [8], DynACL [17], and DeACL [22] on benchmark datasets CIFAR-10 and CIFAR-100 [16]. Additionally, we will compare the Robustness Acquisition Time Cost Ratio (RATCR) and the number of training parameters of PPRP with those of the aforementioned baseline methods to verify the training efficiency of PPRP. Next, to demonstrate the generalizability of PPRP, we will use it to train robustness plugins for pre-trained models from different self-supervised learning methods. Finally, we will conduct some necessary ablation experiments to illustrate the choice of hyperparameters for PPRP.

Experimental settings In all experiments, we follow the settings of the baseline methods and use ResNet18 as the self-supervised pre-training base model. We choose SimCLR [3], BYOL [10], and VICReg [1] as the model pre-training methods, with pre-trained model parameters available for download on solo-learn [5]. For all pre-trained models, PPRP trains the robustness plugin for only 25 epochs using SGD as the optimizer. We generate adversarial perturbations for samples

using a 5-step PGD attack. If not mentioned additionally, in all experiments, the rank of the PPRP plugin was set to 4, at which setting the plugin's parameters were only 1.2% of the base model. Detailed PPRP implementation code and parameter settings can be found in the supplementary materials.

Evaluation criteria During the inference or evaluation phase, we merge the parameters of the robustness plugin with those of the pre-trained base model. The merged parameters form the robust base model capable of producing robust representations. We freeze the parameters of the robust base model after adding the robustness plugin and perform linear fine-tuning using natural samples only. We evaluate the standard accuracy of the robust base model and its robust accuracy under AutoAttack [6].

4.1 Comparing PPRP with state-of-the-arts

Table 1: Comparison of self-AT methods on CIFAR-10 and CIFAR-100. SA and AA stand for standard accuracy and robust accuracy under AutoAttack [6]. The top performance is highlighted in **bold**.

Method	Pretrain-Method	CIFAR-10		CIFAR-100	
		AA (%)	SA (%)	AA (%)	SA (%)
RoCL	SimCLR	26.12	77.90	8.72	42.93
ACL	SimCLR	37.62	79.32	15.68	45.34
AdvCL	SimCLR	37.46	73.23	15.45	37.58
DynACL	SimCLR	45.04	77.41	19.25	45.73
DeACL	SimCLR	45.31	80.17	20.34	52.79
PPRP (ours)	SimCLR	**46.20**	**86.97**	**22.71**	**61.79**

Robustness on various datasets. To maintain consistency with the baseline, we chose SimCLR [3] as the base model pre-training method and used PPRP to train a robust plugin for the base model. As shown in Table 1, the base model with the added PPRP robust plugin exhibits superior standard accuracy and robust accuracy on the CIFAR-10 and CIFAR-100 datasets compared to the baseline method. During the SSL pre-training process, the model's objective is to acquire excellent data representation capabilities. PPRP does not intervene in the base model's pre-training process; rather, its objective is to preserve the base model's effective representation capabilities while enhancing its robustness. PPRP's robustness plugin employs incremental updates to the original base model parameters. Through this way, PPRP achieves the dual goals of maintaining the base model's representation capabilities while improving its robustness.

Robustness on various base model. The integration of adversarial training into model pre-training typically depends on specific self-supervised learning

Table 2: Robustness improvement results of PPRP for different SSL pre-trained base models on CIFAR-10. The top performance is highlighted in **bold**.

Pretrain-Method	DeACL		PPRP (our)	
	AA (%)	SA (%)	AA (%)	SA (%)
SimCLR	45.31	80.17	**46.20**	**86.97**
BYOL	44.14	80.89	**48.77**	**87.37**
VICReg	43.65	82.10	**44.03**	**86.44**

(SSL) methods, which means that these approaches cannot be transferred across different SSL methods. In contrast, PPRP is a versatile model robustness improvement method that can be seamlessly applied to different SSL methods. It provides robust plugins for base models trained with different SSL methods, thus improving the robust representation capabilities of these base models. In Table 2, we trained robust plugins for different base models and evaluated the robustness of the base models with the added robust plugins. As shown in Table 2, the robust plugins trained by PPRP can adapt to base models trained with different SSL methods, and for different base models, PPRP achieves higher standard accuracy and robustness compared to DeACL.

Table 3: Training duration, Robustness Acquisition Time Cost Ratio (RATCR), and total training parameters for different methods. PPRP requires the least training time, has the smallest RATCR, and uses the fewest training parameters.

Method	Training Time (h)	RATCR	Trainable parameters (M)
ACL	32.7	3.057	11.97
AdvCL	105	12.027	12.67
DynACL	30.3	2.759	11.97
DeACL	5.5	0.682	11.31
PPRP (ours)	**0.37**	**0.046**	**0.14**

Training efficiency of PPRP. The training time for PPRP's robust plugin is only 22 minutes, significantly reducing the required training time compared to state-of-the-art (SOTA) methods. We also calculated the Robustness Acquisition Time Cost Ratio (RATCR) for different methods. RATCR indicates the additional time cost required for a model to achieve robustness. When RATCR is greater than 1, it means that the time spent to obtain robustness exceeds the time required for the model's self-supervised pre-training. As shown in Table 3, integrating adversarial training during model pre-training requires 2.75 to 12 times the pre-training time to achieve robustness. Both PPRP and DeACL improve the base model's robustness after the base model pre-training, reducing the time cost of acquiring robustness. However, DeACL requires retraining a new

base model, whereas PPRP only performs incremental updates on the trained base model. Therefore, PPRP converges in fewer training epochs, resulting in an RATCR nearly 15 times lower than that of DeACL. Since PPRP trains a robust plugin with only a small number of parameters, its training parameter count is only 1% of the baseline method, significantly reducing the computational resources required for training a robust model.

4.2 Ablation study for PPRP

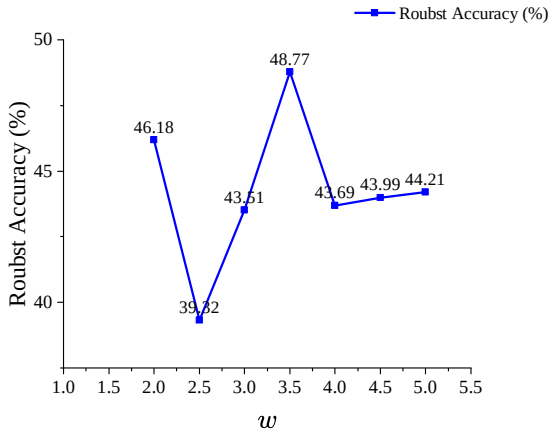

Fig. 3: Effects of weight adjustment parameters of robust plugins on model robustness.

Effect of weight adjustment parameter w. Equation 1 consists of two terms. The first term ensures that the robust plugin does not compromise the base model's representation performance, while the second term uses adversarial training to train the robust plugin, ensuring that the base model with the added robust plugin has excellent robustness. The hyperparameter w in the equation is used to balance the representation performance and robustness of the base model. During training, since the parameters of the robust plugin are initialized to 0, the MSE loss of natural samples in the first term of Equation 1 starts from 0 and gradually increases, while the Mean Square Error (MSE) loss of adversarial samples in the second term of Equation 1 gradually decreases. As shown in Figure 3, on the CIFAR-10 dataset, when $w = 3.5$, the robust plugin provides the highest robustness. In all experiments in this paper, we set w to 3.5.

Effect of different rank choices. In LoRA's low-rank matrix decomposition, different choices of decomposition rank indicate the size of the model's trainable parameters. The higher the set rank, the more trainable parameters the

Table 4: Performance of PPRP with different rank.

r	AA(%)	SA(%)	Trainable parameters (M)
2	44.54	83.33	0.07
4	48.77	87.37	0.14
8	50.71	87.01	0.28
16	50.76	86.23	0.55
32	50.82	88.06	1.05

model has. In PPRP, this corresponds to the size of the robust plugin or the number of training parameters. As shown in Table 5, with the increase in rank (i.e., the size of the robust plugin), the robustness of the base model gradually improves. However, when the rank of the robust plugin is between 8 and 32, the improvement in the base model's robustness starts to plateau, while the size of the robust plugin increases exponentially.

5 Conclusion

This paper introduces the Plug-and-Play Robustness Plugin (PPRP), a training framework for robustness enhancement plugins that significantly boosts the adversarial robustness of pre-trained base models. PPRP trains a plugin with a small number of parameters, and the base model with this added plugin can produce robust representations without increasing inference latency. Experimental results show that PPRP reduces the time required for robustness enhancement to 1/15 of the previously fastest method. The base models with the PPRP robustness plugin achieve state-of-the-art robustness across various datasets, providing an efficient and practical solution for enhancing model security against adversarial attacks. The PPRP framework provides a lightweight adversarial attack defense model plugin for resource-constrained IoT devices in power distribution grids, effectively enhancing the security of data flow. It features high efficiency, low latency, and good scalability, making it suitable for various pre-trained models and heterogeneous devices, meeting the security needs of the diverse devices within power distribution grids.

6 Acknowledgement

This work was supported by the Science and Technology Programme of STATE GRID Corporation of China (5100-202456026A-1-1-ZN).

References

1. Bardes, A., Ponce, J., LeCun, Y.: VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning (Jan 2022). https://doi.org/10.48550/arXiv.2105.04906 1, 3, 6

2. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020) 2
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A Simple Framework for Contrastive Learning of Visual Representations (Jun 2020). https://doi.org/10.48550/arXiv.2002.05709 1, 3, 6, 7
4. Chen, X., He, K.: Exploring Simple Siamese Representation Learning. arXiv:2011.10566 [cs] (Nov 2020) 1, 3
5. da Costa, V.G.T., Fini, E., Nabi, M., Sebe, N., Ricci, E.: solo-learn: A library of self-supervised methods for visual representation learning. Journal of Machine Learning Research **23**(56), 1–6 (2022), http://jmlr.org/papers/v23/21-1155.html 6
6. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks (Aug 2020). https://doi.org/10.48550/arXiv.2003.01690 7
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018) 3
8. Fan, L., Liu, S., Chen, P.Y., Zhang, G., Gan, C.: When Does Contrastive Learning Preserve Adversarial Robustness from Pretraining to Finetuning? (Nov 2021). https://doi.org/10.48550/arXiv.2111.01124 2, 4, 6
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Nets. In: Advances in Neural Information Processing Systems. vol. 27. Curran Associates, Inc. (2014) 2
10. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M.: Bootstrap your own latent: A new approach to self-supervised Learning. arXiv:2006.07733 [cs, stat] (Sep 2020) 1, 3, 6
11. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked Autoencoders Are Scalable Vision Learners (Dec 2021). https://doi.org/10.48550/arXiv.2111.06377 3
12. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020) 1, 3
13. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-Rank Adaptation of Large Language Models (Oct 2021) 2, 5
14. Jiang, Z., Chen, T., Chen, T., Wang, Z.: Robust Pre-Training by Adversarial Contrastive Learning. In: Advances in Neural Information Processing Systems. vol. 33, pp. 16199–16210. Curran Associates, Inc. (2020) 2, 4, 6
15. Kim, M., Tack, J., Hwang, S.J.: Adversarial Self-Supervised Contrastive Learning (Jun 2020). https://doi.org/10.48550/arXiv.2006.07589 2, 4, 6
16. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep. (2009) 6
17. Luo, R., Wang, Y., Wang, Y.: Rethinking the Effect of Data Augmentation in Adversarial Contrastive Learning. https://arxiv.org/abs/2303.01289v2 (Mar 2023) 2, 4, 6
18. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017) 2, 4
19. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018) 2

20. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019) 2, 3
21. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow Twins: Self-Supervised Learning via Redundancy Reduction (Mar 2021). https://doi.org/10.48550/arXiv.2103.03230 3
22. Zhang, C., Zhang, K., Zhang, C., Niu, A., Feng, J., Yoo, C.D., Kweon, I.S.: Decoupled Adversarial Contrastive Learning for Self-supervised Adversarial Robustness (Jul 2022). https://doi.org/10.48550/arXiv.2207.10899 2, 4, 5, 6