Enhancing Adversarial Contrastive Learning via Adversarial Invariant Regularization

Xilie Xu[1]*, **Jingfeng Zhang**[2,3]*†, **Feng Liu**[4], **Masashi Sugiyama**[2,5], **Mohan Kankanhalli**[1]
[1] School of Computing, National University of Singapore
[2] RIKEN Center for Advanced Intelligence Project (AIP)
[3] School of Computer Science, The University of Auckland
[4] School of Computing and Information Systems, The University of Melbourne
[5] Graduate School of Frontier Sciences, The University of Tokyo
xuxilie@comp.nus.edu.sg jingfeng.zhang@auckland.ac.nz
fengliu.ml@gmail.com sugi@k.u-tokyo.ac.jp
mohan@comp.nus.edu.sg

Abstract

Adversarial contrastive learning (ACL) is a technique that enhances standard contrastive learning (SCL) by incorporating adversarial data to learn a robust representation that can withstand adversarial attacks and common corruptions without requiring costly annotations. To improve transferability, the existing work introduced the standard invariant regularization (SIR) to impose *style-independence property* to SCL, which can exempt the impact of nuisance *style factors* in the standard representation. However, it is unclear how the style-independence property benefits ACL-learned robust representations. In this paper, we leverage the technique of *causal reasoning* to interpret the ACL and propose adversarial invariant regularization (AIR) to enforce independence from style factors. We regulate the ACL using both SIR and AIR to output the robust representation. Theoretically, we show that AIR implicitly encourages the representational distance between different views of natural data and their adversarial variants to be independent of style factors. Empirically, our experimental results show that invariant regularization significantly improves the performance of state-of-the-art ACL methods in terms of both standard generalization and robustness on downstream tasks. To the best of our knowledge, we are the first to apply causal reasoning to interpret ACL and develop AIR for enhancing ACL-learned robust representations. Our source code is at https://github.com/GodXuxilie/Enhancing_ACL_via_AIR.

1 Introduction

The attention towards pre-trained models that can be easily finetuned for various downstream applications has significantly increased recently [17, 18, 19, 41]. Notably, foundation models [4] via self-supervision on large-scale unlabeled data, such as GPT [5] and CLAP [20], can be adapted to a wide range of downstream tasks. Due to the high cost of annotating large-scale data, unsupervised learning techniques [21, 34, 49, 16] are commonly used to obtain generalizable representations, in which standard contrastive learning (SCL) has been shown as the most effective way [9, 38, 10].

Adversarial contrastive learning (ACL) [30, 29], that incorporates adversarial data [37] with SCL [9], can yield robust representations that are both generalizable and robust against adversarial attacks [23]

*The first two authors have made equal contributions.

†Corresponding author.

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

https://doi.org/10.52202/075280-0734

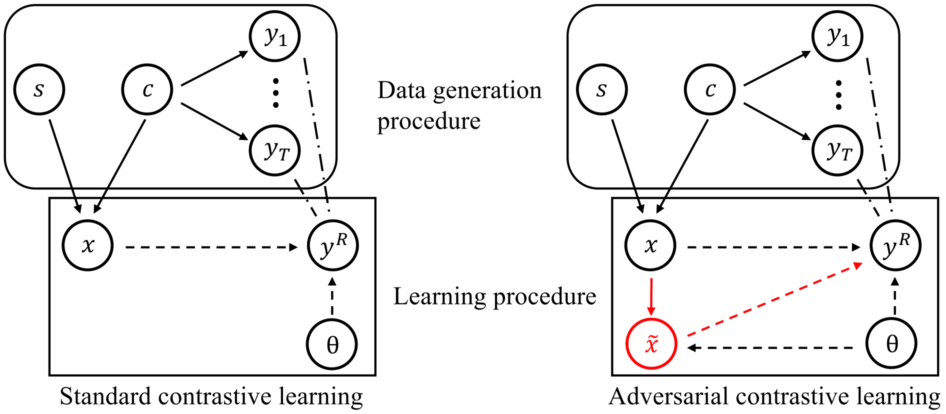
s
c
y_1
y_T
Data generation procedure
x
y^R
θ
Learning procedure
$\tilde{x}$
Standard contrastive learning
Adversarial contrastive learning

Figure 1: Causal graph of data generation procedure in SCL [38] (left panel) and ACL (right panel). s is style variable, c is content variable, x is unlabeled data, $\tilde{x}$ is the generated adversarial data, and θ is the parameter of representation. The solid arrows are causal, but the dash arrows are not causal. The proxy label $y^R \in \mathcal{Y}^R$ is a refinement of the target labels $y_t \in \mathcal{Y} = \{y_i\}_{i=1}^T$ that represents an unknown downstream task. In the causal view, each augmented view of a data point has its own proxy label, where both SCL and ACL aim to minimize the differences between the representation outputs of different views of the same data.

and common corruptions [26]. ACL is attracting increasing popularity since the adversarial robustness of the pre-trained models is essential to safety-critical applications [6, 33]. Many studies have tried to improve ACL from various perspectives including increasing contrastive views and leveraging pseudo labels [22], leveraging hard negative sampling [42, 50], and dynamically scheduling the strength of data augmentations [36].

The style-independence property of learned representations, which eliminates the effects of nuisance style factors in SCL, has been shown to improve the transferability of representations [38]. To achieve style-independence property in learned representations, Mitrovic et al. [38] proposed a standard invariant regularization (SIR) that uses causal reasoning [39, 40] to enforce the representations of natural data to be invariant to style factors. SIR has been shown effective in enhancing representation' transferability. However, it remains unclear in the literature how to achieve style independence in robust representations learned via ACL, especially when adversarial data is used during pre-training.

Therefore, we leverage the technique of causal reasoning to enforce robust representations learned via ACL to be style-independent. As shown in the right panel of Figure 1, we construct the causal graph of ACL (details refer to Section 3.1). Different from the causal graph of SCL [38], ACL has an extra path $x \to \tilde{x}$ since ACL will generate the adversarial data $\tilde{x}$ given the unlabeled data x (i.e., $x \to \tilde{x}$). Then, ACL learns representations by maximizing both the probability of the proxy label y^R given the natural data and that given the adversarial data (i.e., $x \to y^R$ and $\tilde{x} \to y^R$). Theorem 1 shows that maximizing the aforementioned probability in the causal view is equivalent to the learning objective of ACL [29], which justifies the rationality of our constructed causal graph of ACL.

To enforce robust representations to be style-independent, we propose an adversarial invariant regularization (AIR). Specifically, AIR (see Eq. (7)) aims to penalize the Kullback–Leibler divergence between the robust representations of the unlabeled data augmented via two different data augmentation functions during the learning procedure of $x \to \tilde{x} \to y^R$. Note that SIR [38] is a special case of AIR in the standard context where $x = \tilde{x}$. Then, we propose to learn robust representations by minimizing the adversarial contrastive loss [29, 22, 50, 36] together with a weighted sum of SIR and AIR and show the learning algorithm in Algorithm 1.

Furthermore, we give a theoretical understanding of the proposed AIR and show that AIR is beneficial to the robustness against corruptions on downstream tasks. Based on the decomposition of AIR shown in Lemma 2, we propose Proposition 3 which explains that AIR implicitly enforces the representational distance between original as well as augmented views of natural data and their adversarial variants to be independent of the style factors. In addition, we theoretically show that the style-independence property of robust representations learned via ACL will still hold on the downstream classification tasks in Proposition 4, which could be helpful to improve robustness against input perturbations on downstream tasks [3, 45, 28, 38].

Empirically, we conducted comprehensive experiments on various datasets including CIFAR-10 [31], CIFAR-100 [31], STL-10 [12], CIFAR-10-C [26], and CIFAR-100-C [26] to show the effectiveness of our proposed method in improving ACL methods [29, 22, 50, 36]. We demonstrate that our proposed method can achieve the new state-of-the-art (SOTA) performance on various downstream tasks by significantly enhancing standard generalization and robustness against adversarial attacks [13, 14, 2] and common corruptions [26]. Notably, compared with the prior SOTA method DynACL [36], AIR improves the standard and robust test accuracy by 2.71% (from 69.59±0.08% to 72.30±0.10%) and 1.17% (from 46.49±0.05% to 47.66±0.06%) on the STL-10 task and increases the test accuracy under common corruptions by 1.15% (from 65.60±0.18% to 66.75±0.10%) on CIFAR-10-C.

2 Related Works and Preliminaries

Here, we introduce the related works in ACL and causal reasoning and provide the preliminaries.

2.1 Related Works

Adversarial contrastive learning (ACL). Contrastive learning (CL) is frequently used to leverage large unlabeled datasets for learning useful representations. Chen et al. [9] presented SimCLR that leverages contrastive loss for learning useful representations and achieved significantly improved standard test accuracy on downstream tasks. Recently, adversarial contrastive learning (ACL) [30, 29, 27, 22, 50, 52, 36] has become the most effective unsupervised approaches to learn robust representations. Jiang et al. [29], which is the seminal work of ACL that incorporates adversarial data [37] with contrastive loss [9], showed that ACL can exhibit better robustness against adversarial attacks [23, 13] and common corruptions [26] on downstream tasks compared with SCL [9].

One research line focuses on improving the performance of ACL. AdvCL [22] leverages a third contrastive view for high-frequency components and pseudo labels generated by clustering to improve ACL. Yu et al. [50] proposed the asymmetric InfoNCE objective (A-InfoNCE) that utilizes the hard negative sampling method [42] to further enhance AdvCL [22]. Recently, DynACL [36] dynamically schedules the strength of data augmentations and correspondingly schedules the weight for standard and adversarial contrastive losses, achieving the SOTA performance among existing ACL methods. Another research line focuses on improving the efficiency of ACL since ACL requires tremendous running time to generate the adversarial training data during pre-training. Xu et al. [48] proposed a robustness-aware coreset selection (RCS) method to speed up ACL by decreasing the number of training data. Our paper, belonging to the former research line, leveraged the technique of causal reasoning to further enhance ACL and achieved new SOTA results. We treat the incorporation of our proposed method with RCS [48] as future work.

Causal reasoning. Causal reasoning [39, 53, 40, 8, 56, 38, 7, 44, 43] has been widely applied to machine learning to identify causal relations and ignore nuisance factors by intervention. In particular, Zhang et al. [53, 56] investigated the causality in supervised adversarial training [37, 55] where label information is required and proposed the regularizer to eliminate the difference between the natural and adversarial distributions. Mitrovic et al. [38] built the causal graph of SCL in the standard context and introduced the standard invariant regularization (SIR) which aims to enforce the representation of unlabeled natural data to be invariant of the data augmentations. SIR can significantly improve robustness transferability to downstream tasks in terms of common corruptions [26], which is empirically validated by Mitrovic et al. [38].

2.2 Preliminaries

Standard contrastive learning (SCL) [9]. Let $h_\theta : \mathcal{X} \to \mathcal{Z}$ be a representation extractor parameterized by θ, $g : \mathcal{Z} \to \mathcal{V}$ be a projection head that maps representations to the space where the contrastive loss is applied, and $\tau_i, \tau_j : \mathcal{X} \to \mathcal{X}$ be two transformation operations randomly sampled from a pre-defined transformation set $\mathcal{T}$. Given a minibatch $B \sim \mathcal{X}^\beta$ consisting of β original samples, we denote the augmented minibatch $B^u = \{\tau_u(x_k) \mid \forall x_k \in B\}$ via augmentation function $\tau_u(\cdot)$. We take $f_\theta(\cdot) = g \circ h_\theta(\cdot)$ and $x_k^u = \tau_u(x_k)$ for any $x_k \sim \mathcal{X}$ and $u \in \{i, j\}$. Given a positive pair (x_k^i, x_k^j), the standard contrastive loss proposed by SimCLR [9] is formulated as follows:

$$\ell_{\mathrm{CL}}(x_k^i, x_k^j; \theta) = -\sum_{u \in \{i,j\}} \log \frac{e^{\mathrm{sim}\left(f_\theta(x_k^i), f_\theta(x_k^j)\right)/t}}{\sum\limits_{x \in B^i \cup B^j \setminus \{x_k^u\}} e^{\mathrm{sim}\left(f_\theta(x_k^u), f_\theta(x)\right)/t}}, \tag{1}$$

https://doi.org/10.52202/075280-0734

where $\mathrm{sim}(p,q) = p^\top q/\|p\|\|q\|$ is the cosine similarity function and $t > 0$ is a temperature parameter.

ACL [29] and DynACL [36]. Let $(\mathcal{X}, d_\infty)$ be the input space $\mathcal{X}$ with the infinity distance metric $d_\infty(x, x') = \|x - x'\|_\infty$, and $\mathcal{B}_\epsilon[x] = \{x' \in \mathcal{X} \mid d_\infty(x, x') \leq \epsilon\}$ be the closed ball of radius $\epsilon > 0$ centered at $x \in \mathcal{X}$. Given a data point $x_k \in \mathcal{X}$, the adversarial contrastive loss is as follows:

$$\ell_{\mathrm{ACL}}(x_k; \theta) = (1 + \omega) \cdot \ell_{\mathrm{CL}}(\tilde{x}_k^i, \tilde{x}_k^j; \theta) + (1 - \omega) \cdot \ell_{\mathrm{CL}}(x_k^i, x_k^j; \theta), \tag{2}$$

$$\text{where} \quad \tilde{x}_k^i, \tilde{x}_k^j = \mathop{\arg\max}_{\substack{\tilde{x}_k^i \in \mathcal{B}_\epsilon[x_k^i] \\ \tilde{x}_k^j \in \mathcal{B}_\epsilon[x_k^j]}} \ell_{\mathrm{CL}}(\tilde{x}_k^i, \tilde{x}_k^j; \theta), \tag{3}$$

in which $\omega \in [0, 1]$ is a hyperparameter and $\tilde{x}_k^i$ and $\tilde{x}_k^j$ are adversarial data generated via projected gradient descent (PGD) [37, 54, 55] within the ϵ-balls centered at x_k^i and x_k^j. Note that ACL [29] fixes $\omega = 0$ while DynACL [36] dynamically schedules ω according to its dynamic augmentation scheduler that gradually anneals from a strong augmentation to a weak one. We leave the details of the data augmentation scheduler [36] in Appendix C due to the page limitation.

3 Methodology

In this section, we first create the causal graph for ACL which is the fundamental premise for causal reasoning. Then, we introduce adversarial invariant regularization (AIR) according to the causal understanding in the context of ACL and the learning algorithm of our proposed method. Lastly, we demonstrate the theoretical understanding of AIR and theoretically show that the style-independence property is generalizable to downstream tasks.

3.1 Adversarial Contrastive Learning (ACL) in the View of Causality

A causal graph is arguably the fundamental premise for causal reasoning [39, 40]. Previously, Mitrovic et al. [38] constructed the causal graph of SCL as shown in the left panel of Figure 1. However, how to build causal graphs of ACL is still unclear in the literature. Therefore, we take the primitive step to construct the causal graph of ACL.

To begin with, we formalize the data generation procedure in the causal graph. Let $x \in \mathcal{X}$ denote an unlabeled data point and $\mathcal{Y} = \{y_t\}_{t=1}^T$ denote a set of labels in an unknown downstream task. Then, similar to Mitrovic et al. [38] and Zhang et al. [56] we make the following assumptions: 1) the data point x is generated from the content factor c and the style factor s, i.e., $c \rightarrow x \leftarrow s$, 2) the content is independent of the style, i.e., $c \perp\!\!\!\perp s$, and 3) only the content, which is the original image data, is related to the downstream task, i.e., $c \rightarrow y_t$.

Then, we build the learning procedure of CL. According to the above assumptions, the content serves an essential role in downstream classification tasks. Therefore, CL can be regarded as estimating the probability of the label given the content, i.e., $p(y_t|c)$. To learn the representations from the unlabeled data x, CL needs to maximize the conditional probability $p(y_t|x)$.

However, in practice, the label y_t is unknown. Thanks to the causal concept of refinement [8, 38], we can construct a proxy label $y^R \in \mathcal{Y}^R$ which is a refinement of $\mathcal{Y}$. This proxy label y^R should be consistent across different views of the unlabeled data x, which enables learning the representations. Intuitively, the proxy label y^R can be viewed as a more fine-grained variant of the original label y_t. For example, a refinement for recognizing cars would be the task of classifying various types of cars. Therefore, SCL targets maximizing the probability of the proxy label given the unlabeled natural data, i.e., $p(y^R|x)$ [38].

Besides $x \rightarrow y^R$, ACL has an extra path $x \rightarrow \tilde{x} \rightarrow y^R$ since ACL needs to first generate the adversarial data (Eq. (3)) and then learns representations from the unlabeled natural and adversarial data (Eq. (2)). According to Eq. (3), we observe that ACL only uses the observed unlabeled data x for generating adversarial data, i.e., $x \rightarrow \tilde{x} \leftarrow \theta$. Then, according to the paths $x \rightarrow y^R$ and $\tilde{x} \rightarrow y^R$, from the causal view, ACL learns representations by maximizing both the probability of the proxy label y^R given the natural data (i.e., $p(y^R|x)$) and that given adversarial data (i.e., $p(y^R|\tilde{x})$).

Lastly, we provide Theorem 1 to justify the rationality of our constructed causal graph of ACL.

Theorem 1. *From the causal view, in ACL, maximizing the conditional probability both $p(y^R|x)$ and $p(y^R|\tilde{x})$ is equivalent to minimizing the learning objective of ACL [29] that is the sum of standard and adversarial contrastive losses.*

The proof of Theorem 1 is in Appendix B.1.

3.2 Adversarial Invariant Regularization

According to the independence of causal mechanisms [40], performing interventions on the style variable s should not change the probability of the proxy label given the unlabeled data, i.e., $p(y^R|x)$. According to the path $x \rightarrow \tilde{x} \rightarrow y^R$ shown in the causal graph of ACL, we can obtain that

$$p(y^R|x) = p(y^R|\tilde{x})p(\tilde{x}|x), \tag{4}$$

under the assumption that the process of $x \rightarrow \tilde{x} \rightarrow y^R$ satisfies the Markov condition. Therefore, we should make the conditional probability $p(y^R|\tilde{x})p(\tilde{x}|x)$ that is learned via ACL become style-independent. It guides us to propose the following criterion that should be satisfied by ACL:

$$p^{do(\tau_i)}(y^R|\tilde{x})p^{do(\tau_i)}(\tilde{x}|x) = p^{do(\tau_j)}(y^R|\tilde{x})p^{do(\tau_j)}(\tilde{x}|x) \quad \forall \tau_i, \tau_j \in \mathcal{T}, \tag{5}$$

where $do(\tau)$ as the short form of $do(s = \tau)$ denotes that we perform the intervention on s via data augmentation function $\tau(\cdot)$. We leverage the representational distance between various views of natural data and their adversarial variants normalized by the softmax function to calculate the conditional probability $p^{do(\tau_u)}(y^R|\tilde{x})$ and $p^{do(\tau_u)}(\tilde{x}|x)$ where $u \in \{i, j\}$ as follows:

$$p^{do(\tau_u)}(y^R|\tilde{x}) = \frac{e^{\mathrm{sim}(f_\theta(x), f_\theta(\tilde{x}^u))/t}}{\sum\limits_{x_k \in B} e^{\mathrm{sim}\left(f_\theta(x_k), f_\theta(\tilde{x}_k^u)\right)/t}}, \quad p^{do(\tau_u)}(\tilde{x}|x) = \frac{e^{\mathrm{sim}(f_\theta(\tilde{x}^u), f_\theta(x^u))/t}}{\sum\limits_{x_k \in B} e^{\mathrm{sim}\left(f_\theta(\tilde{x}_k^u), f_\theta(x_k^u)\right)/t}}, \tag{6}$$

in which $x \in B \sim \mathcal{X}^\beta$ is the original view of natural data, x^u is the augmented view of natural data via augmentation $\tau_u(\cdot)$, $\tilde{x}^u$ is the adversarial variant generated via Eq. (3), and t is the temperature parameter. Note that we use x as the approximated surrogate of its true content c that incurs y^R (i.e., $c \rightarrow y^R$).

To achieve the above criterion in Eq. (5), we propose an adversarial invariant regularization (AIR) to regulate robust representations as follows:

$$\mathcal{L}_{\mathrm{AIR}}(B; \theta, \epsilon) = \mathrm{KL}\left(p^{do(\tau_i)}(y^R|\tilde{x})p^{do(\tau_i)}(\tilde{x}|x) \| p^{do(\tau_j)}(y^R|\tilde{x})p^{do(\tau_j)}(\tilde{x}|x); B\right), \tag{7}$$

in which $\epsilon \geq 0$ is the adversarial budget and $\mathrm{KL}(p(x)\|q(x); B) = \sum_{x \in B} p(x) \log \frac{p(x)}{q(x)}$ denotes the Kullback–Leibler (KL) divergence. AIR can enforce the representation to satisfy the criterion in Eq. (5), thus eliminating the effect of the style factors on the representation. Therefore, when setting ϵ as a positive constant, $\mathcal{L}_{\mathrm{AIR}}(B; \theta, \epsilon)$ can effectively regulate robust representations of adversarial data to be style-independent.

Besides, to explicitly regulate standard representations of natural data to be independent of style factors, we can simply set $\epsilon = 0$ of AIR. We formulate AIR with $\epsilon = 0$ as follows:

$$\mathcal{L}_{\mathrm{AIR}}(B; \theta, 0) = \mathrm{KL}\left(p^{do(\tau_i)}(y^R|x) \| p^{do(\tau_j)}(y^R|x); B\right), \tag{8}$$

$$\text{where} \quad p^{do(\tau_u)}(y^R|x) = \frac{e^{\mathrm{sim}(f_\theta(x), f_\theta(x^u))/t}}{\sum\limits_{x_k \in B} e^{\mathrm{sim}\left(f_\theta(x_k), f_\theta(x_k^u)\right)/t}} \quad \forall u \in \{i, j\}.$$

Note that $\mathcal{L}_{\mathrm{AIR}}(B; \theta, 0)$ is the same formulation as the standard invariant regularization (SIR) [38], which can make the standard representation style-independent.

By incorporating adversarial contrastive loss with AIR, the learning objective function of our proposed method is shown as follows:

$$\arg\min_\theta \sum_{x \in U} \ell_{\mathrm{ACL}}(x; \theta) + \lambda_1 \cdot \mathcal{L}_{\mathrm{AIR}}(U; \theta, 0) + \lambda_2 \cdot \mathcal{L}_{\mathrm{AIR}}(U; \theta, \epsilon) \tag{9}$$

where $U \sim \mathcal{X}^N$ refers to an unlabeled dataset consisting of N samples, $\epsilon > 0$ is the adversarial budget, and $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ are two hyperparameters. The learning algorithm of ACL with AIR is demonstrated in Algorithm 1. Note that our proposed AIR is compatible with various learning objectives such as ACL [29], AdvCL [22], A-InfoNCE [50], and DynACL [36].

https://doi.org/10.52202/075280-0734

Algorithm 1 ACL with Adversarial Invariant Regularization (AIR)

```
Input: Unlabeled training set $U$, total training epochs $E$, learning rate $\eta$, batch size $\beta$, adversarial
  budget $\epsilon > 0$, hyperparameters $\lambda_1$ and $\lambda_2$
Output: Pre-trained representation extractor $h_\theta$
Initialize parameters of model $f_\theta = g \circ h_\theta$
for $e = 0$ to $E - 1$ do
    for batch $m = 1, \dots, \lceil |U|/\beta \rceil$ do
        Sample a minibatch $B_m$ from $U$
        Update $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \sum_{x_k \in B_m} \ell_{\mathrm{ACL}}(x_k; \theta) + \lambda_1 \cdot \mathcal{L}_{\mathrm{AIR}}(B_m; \theta, 0) + \lambda_2 \cdot \mathcal{L}_{\mathrm{AIR}}(B_m; \theta, \epsilon)$
    end for
end for
```

3.3 Theoretical Analysis

We provide a theoretical understanding of AIR in Proposition 3 based on the decomposition of AIR shown in Lemma 2. We set $\epsilon > 0$ of AIR by default in this section.

Lemma 2 (Decomposition of AIR). *AIR in Eq.* (7) *can be decomposed into two terms as follows:*

$$\begin{aligned}\mathcal{L}_{\mathrm{AIR}}(B; \theta, \epsilon) =& \mathbb{E}_{x \sim p^{do(\tau_i)}(\tilde{x}|x)}[\mathrm{KL}(p^{do(\tau_i)}(y^R|\tilde{x}) \| p^{do(\tau_j)}(y^R|\tilde{x}))] \\ &+ \mathbb{E}_{x \sim p^{do(\tau_i)}(y^R|\tilde{x})}[\mathrm{KL}(p^{do(\tau_i)}(\tilde{x}|x) \| p^{do(\tau_j)}(\tilde{x}|x))],\end{aligned}$$

where $\mathbb{E}_{x \sim Q^3(x)}[\mathrm{KL}(Q^1(x) \| Q^2(x))]$ *is the expectation of KL divergence over* Q^3 *and* Q^1, Q^2, Q^3 *are probability distributions.*

The proof of Lemma 2 is in Appendix B.2. Based on Lemma 2, we propose Proposition 3 that provides the theoretical understanding of AIR.

Proposition 3. *AIR implicitly enforces the robust representation to satisfy the following two proxy criteria:*

$$(1) \quad p^{do(\tau_i)}(y^R|\tilde{x}) = p^{do(\tau_j)}(y^R|\tilde{x}), \quad (2) \quad p^{do(\tau_i)}(\tilde{x}|x) = p^{do(\tau_j)}(\tilde{x}|x).$$

The proof of Proposition 3 is in Appendix B.3.

Remarks. Proposition 3 explains that AIR implicitly enforces the representational distance to be style-independent between the original view of natural data and their adversarial variants (i.e., $p^{do(\tau_i)}(y^R|\tilde{x}) = p^{do(\tau_j)}(y^R|\tilde{x})$), as well as between the augmented view of natural data and their adversarial counterparts (i.e., $p^{do(\tau_i)}(\tilde{x}|x) = p^{do(\tau_j)}(\tilde{x}|x)$). Therefore, Proposition 3 explicitly presents that AIR is different from SIR [38], where SIR only enforces the augmented views of natural data to be style-independent.

Next, we theoretically show that on the assumption that $\mathcal{Y}^R$ is a refinement of $\mathcal{Y}$, the style-independence property of robust representations will hold on downstream classification tasks in Proposition 4.

Proposition 4. *Let* $\mathcal{Y} = \{y_t\}_{t=1}^T$ *be a label set of a downstream classification task,* $\mathcal{Y}^R$ *be a refinement of* $\mathcal{Y}$*, and* $\tilde{x}_t$ *be the adversarial data generated on the downstream task. Assuming that* $\tilde{x} \in \mathcal{B}_\epsilon[x]$ *and* $\tilde{x}_t \in \mathcal{B}_\epsilon[x]$*, we have the following results:*

$$\begin{aligned}p^{do(\tau_i)}(y^R|\tilde{x}) = p^{do(\tau_j)}(y^R|\tilde{x}) &\Longrightarrow p^{do(\tau_i)}(y_t|\tilde{x}_t) = p^{do(\tau_j)}(y_t|\tilde{x}_t) \quad \forall \tau_i, \tau_j \in \mathcal{T}, \\ p^{do(\tau_i)}(\tilde{x}|x) = p^{do(\tau_j)}(\tilde{x}|x) &\Longrightarrow p^{do(\tau_i)}(\tilde{x}_t|x) = p^{do(\tau_j)}(\tilde{x}_t|x) \quad \forall \tau_i, \tau_j \in \mathcal{T}.\end{aligned}$$

The proof of Proposition 4 is shown in Appendix B.4.

Remarks. Proposition 4 indicates that the robust representation of data on downstream tasks is still style-independent. Mitrovic et al. [38] empirically showed that the style-independence property of standard representations can significantly improve the robustness against common corruptions [26] on downstream tasks. In addition, Proposition 4 indicates that AIR helps to find the invariant correlations that are independent of the style factors among different training distributions. It is similar to the objective of invariant risk minimization [3] that can yield substantial improvement in the robustness against various corruptions incurred by distribution shifts. Therefore, the style-independence property of robust representations learned via ACL perhaps helps to enhance the robustness against input perturbations on downstream tasks.

Table 1: Self-task adversarial robustness transferability.

Dataset	Pre-training	SLF		ALF		AFF	
		AA (%)	SA (%)	AA (%)	SA (%)	AA (%)	SA (%)
CIFAR-10	ACL [29]	37.39±0.06	78.27±0.09	40.61±0.07	75.56±0.09	49.42±0.07	82.14±0.18
	ACL-AIR	**38.89**±0.06	**80.03**±0.07	**41.39**±0.08	**78.29**±0.10	**49.84**±0.04	**82.42**±0.06
	DynACL [36]	45.05±0.04	75.39±0.05	45.65±0.05	72.90±0.08	50.52±0.05	81.86±0.11
	DynACL-AIR	**45.17**±0.04	**78.08**±0.06	**46.01**±0.07	**77.42**±0.10	**50.60**±0.08	**82.14**±0.11
CIFAR-100	ACL [29]	15.78±0.05	45.70±0.10	17.36±0.16	42.69±0.13	24.16±0.29	56.68±0.14
	ACL-AIR	**16.14**±0.07	**49.75**±0.10	**18.68**±0.11	**47.07**±0.15	**25.27**±0.10	**57.79**±0.18
	DynACL [36]	19.31±0.06	45.67±0.09	20.30±0.08	43.58±0.12	24.70±0.23	57.22±0.28
	DynACL-AIR	**20.45**±0.07	**46.84**±0.12	**21.23**±0.09	**45.63**±0.10	**25.34**±0.12	**57.44**±0.14
STL-10	ACL [29]	35.80±0.06	67.90±0.09	38.10±0.11	69.96±0.14	43.21±0.16	72.55±0.18
	ACL-AIR	**36.94**±0.06	**68.91**±0.07	**39.05**±0.10	**71.30**±0.12	**43.75**±0.13	**72.84**±0.14
	DynACL [36]	46.49±0.05	69.59±0.08	47.69±0.10	67.65±0.12	45.64±0.13	72.14±0.15
	DynACL-AIR	**47.66**±0.06	**72.30**±0.10	**48.89**±0.08	**71.60**±0.09	**48.10**±0.11	**73.10**±0.17

Table 2: Self-task common-corruption robustness transferability. The corruption severity (CS) ranging from $\{1, 3, 5\}$ (denoted as "CS-{1,3,5}"). The standard deviation is in Table 15.

Dataset	Pre-training	SLF			ALF			AFF		
		CS-1	CS-3	CS-5	CS-1	CS-3	CS-5	CS-1	CS-3	CS-5
CIFAR-10	ACL [29]	76.57	71.78	62.78	74.04	69.49	61.38	79.15	72.54	65.27
	ACL-AIR	**78.55**	**73.33**	**64.28**	**76.65**	**71.38**	**63.17**	**79.49**	**72.95**	**65.37**
	DynACL [36]	73.92	69.01	62.51	71.74	66.95	60.87	79.77	72.95	65.60
	DynACL-AIR	**76.62**	**70.16**	**63.29**	**75.70**	**69.55**	**62.67**	**80.98**	**74.31**	**66.75**
CIFAR-100	ACL [29]	43.14	38.90	32.27	39.52	35.13	29.30	53.80	45.93	37.61
	ACL-AIR	**47.85**	**41.54**	**34.12**	**44.80**	**40.70**	**35.46**	**55.12**	**47.25**	**39.02**
	DynACL [36]	44.19	38.08	31.05	39.83	35.61	30.51	54.34	46.71	38.62
	DynACL-AIR	**45.36**	**39.21**	**32.44**	**42.03**	**36.48**	**30.49**	**55.31**	**47.33**	**39.13**

4 Experiments

In this section, we demonstrate the effectiveness of our proposed AIR in improving ACL [29] and its variants [36, 22, 50] on various datasets including CIFAR-10 [31], CIFAR-100 [31], STL-10 [12], CIFAR-10-C [26], and CIFAR-100-C [26]. Extensive experimental details are shown in Appendix C.

Pre-training. In the main paper, we demonstrate the results of applying our proposed AIR with $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$ to ACL [29] and DynACL [36] (denoted as "ACL-AIR" and "DynACL-AIR", respectively). We utilized ResNet-18 [25] as the representation extractor following previous self-supervised adversarial training methods [29, 22, 36, 50]. We adopted the same training configuration of ACL [29] using SGD for 1000 epochs with an initial learning rate of 5.0 and a cosine annealing schedule [35]. The batch size β is fixed as 512. The adversarial budget ϵ is set as $8/255$. In the context of DynACL, we took the same data augmentation scheduler and the same scheduler of the hyperparameter ω as the setting in the original paper of DynACL [36]. Note that, to reproduce the results of baseline methods, we downloaded the pre-trained weights of ACL [29] on CIFAR-10/CIFAR-100 and DynACL [36] on CIFAR-10/CIFAR-100/STL-10 from their official GitHub as the pre-trained representation extractor. We provide the extensive ablation studies on the hyper-parameters (λ_1 and λ_2) in Appendix C.2, the ACL variants (including AdvCL [22] and A-InfoNCE [50]) in Appendix C.3, and the backbone models (including ResNet-50 and Wide ResNet [51]) in Appendix C.4.

Finetuning procedures. We adopted the following three types of finetuning procedures: standard linear finetuning (SLF), adversarial linear finetuning (ALF), and adversarial full finetuning (AFF), to evaluate the learned representations. The former two finetuning procedures freeze the learned representation extractor and only train the linear classifier using natural or adversarial samples, respectively. We took the pre-trained representation extractor as weight initialization and trained the whole model using the adversarial data during AFF. The training configuration of finetuning (e.g., the finetuning epoch and optimizer) exactly follows DynACL [36]. Specifically, we used the official code provided in DynACL [36]'s GitHub for finetuning and illustrate the experimental settings of finetuning in Appendix C as well. For each pre-trained representation extractor, we repeated the finetuning experiments 3 times and report the median results and the standard deviation.

Evaluation protocols. We let "AA" denote the robust test accuracy under AutoAttack (AA) [13] and "SA" stand for the standard test accuracy. To evaluate the robustness against common corruption, we report the mean test accuracy under 15 types of common corruptions [26] with the corruption

Table 3: Cross-task adversarial robustness transferability. $\mathcal{D}_1 \rightarrow \mathcal{D}_2$ denotes pre-training and finetuning are conducted on the dataset $\mathcal{D}_1$ and $\mathcal{D}_2 (\neq \mathcal{D}_1)$, respectively.

$\mathcal{D}_1 \rightarrow \mathcal{D}_2$	Pre-training	SLF		ALF		AFF	
		AA (%)	SA (%)	AA (%)	SA (%)	AA (%)	SA (%)
CIFAR-10 → CIFAR-100	ACL [29]	9.98±0.02	32.61±0.04	11.09±0.06	28.58±0.06	22.67±0.16	56.05±0.19
	ACL-AIR	**11.04**±0.06	**39.45**±0.07	**13.30**±0.02	**36.10**±0.05	**23.45**±0.04	**56.31**±0.06
	DynACL [36]	11.01±0.02	27.66±0.03	11.92±0.05	24.14±0.09	24.17±0.10	55.61±0.17
	DynACL-AIR	**12.20**±0.04	**31.33**±0.03	**12.70**±0.03	**28.70**±0.05	**24.82**±0.07	**57.00**±0.13
CIFAR-10 → STL-10	ACL [29]	25.41±0.08	56.53±0.10	27.17±0.09	51.71±0.17	32.66±0.07	61.41±0.13
	ACL-AIR	**28.00**±0.12	**61.91**±0.13	**30.06**±0.10	**62.03**±0.11	**34.26**±0.09	**62.58**±0.10
	DynACL [36]	28.52±0.09	52.45±0.10	29.13±0.13	49.53±0.17	35.25±0.15	63.29±0.18
	DynACL-AIR	**29.88**±0.04	**54.59**±0.12	**31.24**±0.06	**57.14**±0.09	**35.66**±0.05	**63.74**±0.12
CIFAR-100 → CIFAR-10	ACL [29]	18.72±0.07	60.90±0.02	26.92±0.11	57.35±0.07	44.07±0.11	75.19±0.10
	ACL-AIR	**19.90**±0.04	**64.89**±0.09	**27.65**±0.06	**60.79**±0.04	**44.84**±0.14	**75.67**±0.13
	DynACL [36]	25.23±0.12	59.12±0.10	28.92±0.10	56.09±0.14	47.40±0.23	77.92±0.18
	DynACL-AIR	**25.63**±0.07	**59.83**±0.08	**29.32**±0.06	**56.65**±0.06	**47.92**±0.12	**78.44**±0.10
CIFAR-100 → STL-10	ACL [29]	21.77±0.07	46.19±0.05	24.46±0.09	45.40±0.12	28.76±0.07	56.16±0.13
	ACL-AIR	**22.44**±0.04	**51.52**±0.02	**26.55**±0.06	**53.24**±0.09	**30.40**±0.08	**58.45**±0.11
	DynACL [36]	23.17±0.09	47.54±0.14	26.24±0.13	45.70±0.14	31.17±0.14	58.35±0.18
	DynACL-AIR	**23.24**±0.07	**48.20**±0.08	**26.60**±0.05	**48.55**±0.12	**31.42**±0.07	**58.59**±0.10

Table 4: Cross-task common-corruption robustness transferability. The corruption severity (CS) ranging from $\{1, 3, 5\}$ (denoted as "CS-{1,3,5}"). The standard deviation is in Table 16.

$\mathcal{D}_1 \rightarrow \mathcal{D}_2$	Pre-training	SLF			ALF			AFF		
		CS-1	CS-3	CS-5	CS-1	CS-3	CS-5	CS-1	CS-3	CS-5
CIFAR-10 → CIFAR-100	ACL [29]	31.39	27.76	23.27	27.80	25.09	21.09	52.07	44.22	36.31
	ACL-AIR	**37.82**	**32.83**	**27.06**	**34.67**	**30.09**	**25.07**	**52.81**	**44.95**	**37.01**
	DynACL [36]	26.74	23.97	20.87	23.70	21.43	18.84	52.87	44.87	36.76
	DynACL-AIR	**30.35**	**26.53**	**22.76**	**27.67**	**24.35**	**20.96**	**54.00**	**46.01**	**37.75**
CIFAR-100 → CIFAR-10	ACL [29]	59.65	55.14	49.09	56.15	51.96	47.04	72.94	66.05	59.17
	ACL-AIR	**63.34**	**58.14**	**51.03**	**59.14**	**54.23**	**48.97**	**73.48**	**66.83**	**60.21**
	DynACL [36]	57.14	52.07	47.03	56.31	52.03	47.40	75.89	69.08	62.02
	DynACL-AIR	**58.41**	**53.09**	**47.94**	**54.89**	**49.84**	**45.08**	**76.35**	**69.49**	**62.44**

severity (CS) ranging from $\{1, 3, 5\}$ (denoted as "CS-$\{1, 3, 5\}$"). Specifically, we used the official code of AutoAttack [13] and RobustBench [15] for implementing evaluations. In Appendix C.5, we provide robustness evaluation under more diverse attacks [13, 14, 2]. In Appendix C.6, we provide the test accuracy under each type of common corruption [26].

4.1 Evaluation of Self-Task Robustness Transferability

Self-task adversarial robustness transferability. Table 1 reports the self-task adversarial robustness transferability evaluated on three datasets where pre-training and finetuning are conducted on the same datasets including CIFAR-10, CIFAR-100, and STL-10. In Appendix C.1, we report the self-task robustness transferability evaluated in the semi-supervised settings. Table 1 demonstrates that our proposed AIR can obtain new state-of-the-art (SOTA) both standard and robust test accuracy compared with existing ACL methods [29, 36]. Therefore, It validates the effectiveness of our proposed AIR in consistently improving the self-task adversarial robustness transferability against adversarial attacks as well as standard generalization in downstream datasets via various finetuning procedures. Notably, compared with the previous SOTA method DynACL [36], DynACL-AIR increases standard test accuracy by 4.52% (from 72.90% to 77.42%) on the CIFAR-10 dataset via ALF, 2.05% (from 43.58% to 45.63%) on the CIFAR-100 dataset via ALF, and 1.17% (from 46.49% to 47.66%) on the STL-10 dataset via SLF.

Self-task common-corruption robustness transferability. Table 2 reports the self-task common-corruption [26] robustness transferability. Specifically, we conducted pre-training on CIFAR-10/CIFAR-100 and then evaluated the test accuracy on CIFAR-10-C/CIFAR-100-C with various corruption severities after various finetuning procedures. Note that the test accuracy under each type of common corruption is reported in Appendix C.6. Table 2 demonstrates that AIR leads to consistent and significant improvement in the robustness against common corruption. In addition, we observe that ACL always achieves much higher test accuracy under common corruptions than DynACL after SLF. We conjecture the reason is that DynACL uses weak data augmentations at the later phase of training, which makes DynACL more likely to overfit the training distribution of CIFAR-10 and thus leads to worse performance under common corruptions via SLF.

https://doi.org/10.52202/075280-0734

Table 5: Self-task adversarial robustness transferability with post-processing. "++" denotes that the pre-trained models are post-processed via LP-AFF [32, 36]. LP-AFF [32] first generates pseudo labels via clustering and then adversarially pre-trains the model using the pseudo labels.

Dataset	Pre-training	SLF		ALF		AFF	
		AA (%)	SA (%)	AA (%)	SA (%)	AA (%)	SA (%)
CIFAR-10	DynACL++ [36]	46.46†	79.81†	47.95†	78.84†	50.31†	81.94†
	DynACL-AIR++	**46.99**	**81.80**	**48.23**	**79.56**	**50.65**	**82.36**
CIFAR-100	DynACL++ [36]	20.07†	52.26†	22.24	49.92	25.21	57.30
	DynACL-AIR++	**20.61**	**53.93**	**22.96**	**52.09**	**25.48**	**57.57**
STL-10	DynACL++ [36]	47.21†	70.93†	48.06	69.51	41.84	72.36
	DynACL-AIR++	**47.90**	**71.44**	**48.59**	**71.45**	**44.09**	**72.42**

Table 6: Self-task adversarial robustness transferability evaluated on the Imagenette dataset.

Pre-training	ResNet-18		ResNet-50	
	AA (%)	SA (%)	AA (%)	SA (%)
DynACL [36]	57.15	79.41	58.98	80.74
DynACL-AIR	**58.34**	**80.61**	**60.10**	**81.66**

4.2 Evaluation of Cross-Task Robustness Transferability

Cross-task adversarial robustness transferability. Table 3 shows the cross-task adversarial robustness transferability where pre-training and finetuning are conducted on the different datasets. We can observe that AIR substantially improves both ACL's and DynACL's robustness against adversarial attacks [13] and standard generalization to other downstream datasets. Particularly, AIR improves the standard and robust test accuracy of ACL via ALF by 7.52% (from 28.58% to 36.10%) and 2.21% (from 11.09% to 13.30%), respectively.

Cross-task common-corruption robustness transferability. Table 4 reports the cross-task common-corruption [26] robustness transferability where pre-training is conducted on CIFAR-10/CIFAR-100 and finetuning is conducted on CIFAR-100-C/CIFAR-10-C [26]. The empirical results show that our proposed AIR can consistently improve the accuracy under common corruptions with various severity, which validates the effectiveness of AIR in enhancing the robustness transferability against common corruptions.

4.3 Evaluation of Pre-Trained Models after Post-Processing

Here, we report the self-task adversarial robustness transferability of pre-trained models after post-processing in Table 5. The post-processing method is linear probing and then adversarial full finetuning (LP-AFF) [32] which first generates pseudo labels via clustering and then further adversarially trains the pre-trained model using the pseudo labels. "++" denotes that the pre-trained model is post-processed via LP-AFF. We provide the comparison between DynACL++ and DynACL-AIR++ in Table 5. Note that the superscript † denotes that the results of DynACL++ are copied from Luo et al. [36]. The results validate that AIR is compatible with the trick of post-processing via LP-AFF [32], which can yield improved performance on various downstream datasets.

4.4 Evaluation on High-Resolution Imagenette Dataset

To the best of our knowledge, there exists no reported result of ACL and DynACL on high-resolution datasets in the existing papers. Therefore, we pre-trained ResNet-18 and ResNet-50 on Imagenette[3] of resolution 256×256 which is a subset of ImageNet-1K using DynACL and DynACL-AIR, respectively. We compare and report the performance evaluated on Imagenette via SLF in Table 6. Our results validate that our proposed AIR is effective in enhancing the performance on high-resolution datasets. Note that, due to the limitation of our GPU memory, we set the batch size to 256 and 128 for pre-training ResNet-18 and ResNet-50 on Imagenette, respectively. We believe using a larger batch size during pre-training can even further improve the performance [9].

4.5 Evaluation via Automated Robust Fine-Tuning

While evaluating the transferability of a pre-trained model, we observed that the performance on the downstream dataset is sensitive to the hyper-parameters (e.g., the learning rate) of the finetuning

[3] We downloaded the Imagenette dataset from https://github.com/fastai/imagenette.

Table 7: Cross-task adversarial robustness transferability evaluated via AutoLoRa [47]. $\mathcal{D}_1 \rightarrow \mathcal{D}_2$ denotes pre-training and finetuning are conducted on the dataset $\mathcal{D}_1$ and $\mathcal{D}_2(\neq \mathcal{D}_1)$, respectively. “Diff” refers to the gap between the performance achieved by AutoLoRa and that achieved by vanilla finetuning (reported in Table 3).

$\mathcal{D}_1 \rightarrow \mathcal{D}_2$	Finetuning mode	Pre-training	AutoLoRa [47]		Diff	
			AA (%)	SA (%)	AA (%)	SA (%)
CIFAR-10 → STL-10	SLF	DynACL [36]	29.68	58.24	+0.51	+5.83
		DynACL-AIR	**29.75**	**60.53**	**+0.11**	**+4.69**
	ALF	DynACL [36]	31.34	57.74	+1.75	+8.19
		DynACL-AIR	**31.65**	**59.44**	**+0.41**	**+2.30**
	AFF	DynACL [36]	35.55	64.16	+0.30	+0.63
		DynACL-AIR	**35.81**	**64.28**	**+0.15**	**+0.54**
CIFAR-100 → STL-10	SLF	DynACL [36]	23.31	50.93	+0.14	+3.39
		DynACL-AIR	**23.49**	**51.28**	**+0.25**	**+3.08**
	ALF	DynACL [36]	26.53	51.74	+0.29	+6.04
		DynACL-AIR	**26.89**	**49.02**	**+0.29**	**+0.47**
	AFF	DynACL [36]	31.25	58.44	+0.08	+0.09
		DynACL-AIR	**31.57**	**58.65**	**+0.15**	**+0.21**

procedure. It would require extensive computational resources to search for appropriate hyper-parameters to achieve a satisfactory performance. To mitigate this issue, we leverage an automated robust finetuning framework called AutoLoRa [47], which can automatically schedule the learning rate and set appropriate hyper-parameters during finetuning. We report the performance achieved by AutoLoRa to further justify the SOTA performance by our proposed AIR in Table 7.

Table 7 shows that AutoLoRa can further enhance the performance of a pre-trained model on the downstream tasks without searching for appropriate hyper-parameters since the value of “Diff” is consistently larger than 0.0. Besides, Table 7 justifies that our proposed AIR is effective in enhancing robustness transferability via various finetuning methods.

5 Conclusions

This paper leveraged the technique of causal reasoning to interpret ACL and proposed adversarial invariant regularization (AIR) to enhance ACL. AIR can enforce the learned robust representations to be invariant of the style factors. We improved ACL by incorporating the adversarial contrastive loss with a weighted sum of AIR and SIR that is an extension of AIR in the standard context. Theoretically, we showed that AIR implicitly encourages the representational distance between different views of natural data and their adversarial variants to be independent of style factors. Empirically, comprehensive results validate that our proposed method can achieve new state-of-the-art performance in terms of standard generalization and robustness against adversarial attacks and common corruptions on downstream tasks.

Acknowledgements

This research is supported by the National Research Foundation, Singapore under its Strategic Capability Research Centres Funding Initiative, Australian Research Council (ARC) under Award No. DP230101540 and NSF and CSIRO Responsible AI Program under Award No. 2303037. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

[1] Jean-Baptiste Alayrac, Jonathan Uesato, Po-Sen Huang, Alhussein Fawzi, Robert Stanforth, and Pushmeet Kohli. Are labels required for improving adversarial robustness? *Advances in Neural Information Processing Systems*, 32, 2019.

[2] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII*, pages 484–501. Springer, 2020.

[3] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[4] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[6] Varun H Buch, Irfan Ahmed, and Mahiben Maruthappu. Artificial intelligence in medicine: current trends and future possibilities. *British Journal of General Practice*, 68(668):143–144, 2018.

[7] Peter Bühlmann. Invariance, causality and robustness. 2020.

[8] Krzysztof Chalupka, Pietro Perona, and Frederick Eberhardt. Visual causal feature learning. *arXiv preprint arXiv:1412.2309*, 2014.

[9] 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.

[10] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

[11] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021.

[12] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[13] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020.

[14] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pages 2196–2205. PMLR, 2020.

[15] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

[16] Zhiyuan Dang, Cheng Deng, Xu Yang, Kun Wei, and Heng Huang. Nearest neighbor matching for deep clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13693–13702, 2021.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[18] Jiahua Dong, Yang Cong, Gan Sun, Bineng Zhong, and Xiaowei Xu. What can be transferred: Unsupervised domain adaptation for endoscopic lesions segmentation. In *CVPR*, 2020.

[19] Jiahua Dong, Yang Cong, Gan Sun, Zhen Fang, and Zhengming Ding. Where and how to transfer: Knowledge aggregation-induced transferability perception for unsupervised domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):1–18, 2021. doi: 10.1109/TPAMI.2021.3128560.

[20] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap: Learning audio concepts from natural language supervision. *arXiv preprint arXiv:2206.04769*, 2022.

[21] Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. Why does unsupervised pre-training help deep learning? In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 201–208. JMLR Workshop and Conference Proceedings, 2010.

[22] Lijie Fan, Sijia Liu, Pin-Yu Chen, Gaoyuan Zhang, and Chuang Gan. When does contrastive learning preserve adversarial robustness from pretraining to finetuning? *Advances in Neural Information Processing Systems*, 34:21480–21492, 2021.

[23] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *The Third International Conference on Learning Representations*, 2015.

[24] 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.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[26] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

[27] Chih-Hui Ho and Nuno Nvasconcelos. Contrastive learning with adversarial examples. *Advances in Neural Information Processing Systems*, 33:17081–17093, 2020.

[28] Tianjian Huang, Shaunak Ashish Halbe, Chinnadhurai Sankar, Pooyan Amini, Satwik Kottur, Alborz Geramifard, Meisam Razaviyayn, and Ahmad Beirami. Robustness through data augmentation loss consistency. *Transactions on Machine Learning Research*.

[29] Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust pre-training by adversarial contrastive learning. *Advances in Neural Information Processing Systems*, 33:16199–16210, 2020.

[30] Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:2983–2994, 2020.

[31] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[32] Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *The Tenth International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=UYneFzXSJWh`.

[33] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.

[34] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934, 2020.

[35] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[36] Rundong Luo, Yifei Wang, and Yisen Wang. Rethinking the effect of data augmentation in adversarial contrastive learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=0qmwFNJyxCL`.

[37] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[38] Jovana Mitrovic, Brian McWilliams, Jacob C Walker, Lars Holger Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=9p2ekP904Rs`.

[39] Judea Pearl. *Causality*. Cambridge university press, 2009.

[40] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[41] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021.

https://doi.org/10.52202/075280-0734

[42] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020.

[43] Axel Sauer and Andreas Geiger. Counterfactual generative networks. *arXiv preprint arXiv:2101.06046*, 2021.

[44] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *Advances in Neural Information Processing Systems*, 33:1513–1524, 2020.

[45] Ruoyu Wang, Mingyang Yi, Zhitang Chen, and Shengyu Zhu. Out-of-distribution generalization with causal invariant transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 375–385, 2022.

[46] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 819–828, 2020.

[47] Xilie Xu, Jingfeng Zhang, and Mohan Kankanhalli. Autolora: A parameter-free automated robust fine-tuning framework. *arXiv preprint arXiv:2310.01818*, 2023.

[48] Xilie Xu, Jingfeng Zhang, Feng Liu, Masashi Sugiyama, and Mohan Kankanhalli. Efficient adversarial contrastive learning via robustness-aware coreset selection. In *NeurIPS*, 2023.

[49] Xu Yang, Cheng Deng, Kun Wei, Junchi Yan, and Wei Liu. Adversarial learning for robust deep clustering. *Advances in Neural Information Processing Systems*, 33:9098–9108, 2020.

[50] Qiying Yu, Jieming Lou, Xianyuan Zhan, Qizhang Li, Wangmeng Zuo, Yang Liu, and Jingjing Liu. Adversarial contrastive learning via asymmetric infonce. In *European Conference on Computer Vision*, pages 53–69. Springer, 2022.

[51] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[52] Chaoning Zhang, Kang Zhang, Chenshuang Zhang, Axi Niu, Jiu Feng, Chang D Yoo, and In So Kweon. Decoupled adversarial contrastive learning for self-supervised adversarial robustness. In *European Conference on Computer Vision*, pages 725–742. Springer, 2022.

[53] Cheng Zhang, Kun Zhang, and Yingzhen Li. A causal view on robustness of neural networks. *Advances in Neural Information Processing Systems*, 33:289–301, 2020.

[54] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

[55] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan S. Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International Conference on Machine Learning*, pages 11278–11287. PMLR, 2020.

[56] Yonggang Zhang, Mingming Gong, Tongliang Liu, Gang Niu, Xinmei Tian, Bo Han, Bernhard Schölkopf, and Kun Zhang. Adversarial robustness through the lens of causality. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=cZAi1yWpiXQ`.

https://doi.org/10.52202/075280-0734

A Mathematical Notations

Table 8: Notation Table

Notation	Description
$\mathcal{X}$	input space
$x \in \mathcal{X}$	data point
$\mathcal{T}$	transformations set
$\tau \in \mathcal{T}$	data augmentation fraction
x^u	augmented data point via the data augmentation fraction $\tau_u(\cdot)$
$\tilde{x}$	adversarial data
$B \sim \mathcal{X}^\beta$	a minibatch of β original image samples
B^u	a minibatch of β augmented image samples via the data augmentation fraction $\tau_u(\cdot)$
$\tilde{B}^u$	a minibatch of adversarial counterparts of β augmented samples via the data augmentation fraction $\tau_u(\cdot)$
$U \sim \mathcal{X}^N$	an unlabeled data set of N image samples
$\mathcal{Y}$	label space on the downstream tasks
$y_t \in \mathcal{Y}$	target label on the downstream tasks
$\mathcal{Y}^R$	the refinement of the target label space
$y^R \in \mathcal{Y}^R$	the proxy label
$y^R_{ku} \in \mathcal{Y}^R$	the proxy label of the data point x^u_k augmented via $\tau_u(\cdot)$
$\mathcal{Z}$	projection space
$\mathcal{V}$	feature space where the contrastive loss is applied
$h_\theta : \mathcal{X} \to \mathcal{Z}$	the representation extractor parameterized by θ
$g : \mathcal{Z} \to \mathcal{V}$	the projector
$f(\cdot) = g \circ h_\theta(\cdot)$	the composition of the representation extractor and the projector
$\mathrm{sim}(\cdot, \cdot)$	the cosine similarity function
$\mathrm{KL}(\cdot \Vert \cdot)$	KL divergence function
s	style variable
c	content variable
λ_1	the weight of standard regularization
λ_2	the weight of adversarial regularization

B Proof

B.1 Proof of Theorem 1

Theorem 1 (restated). *From the causal view, in ACL, maximizing the conditional probability both $p(y^R|x)$ and $p(y^R|\tilde{x})$ is equivalent to minimizing the learning objective of ACL [29] that is the sum of standard and adversarial contrastive losses.*

Proof. To begin with, we formulate the proxy label driven y^R by the data augmentation in the context of contrastive learning. We denote the index of $x^i_k \in B^i$ and $x^j_k \in B^j$ as ki and kj, respectively. For the augmented data $x^i_k \in B^i$ and $x^j_k \in B^j$, we denote their corresponding proxy labels as y^R_{ki} and y^R_{kj}, which is formulated as follows:

$$y^R_{ki} = \mathbf{1}_{kj}, \quad y^R_{kj} = \mathbf{1}_{ki}, \tag{10}$$

where $\mathbf{1}_{ku} \in \{0, 1\}^{2\beta - 1}$ is a one-hot label where $u \in \{i, j\}$, the value at index ku is 1, and the values at other indexes are 0. The one-hot label refers to the refined proxy label of the augmented view of the data point, which directs to its peer view. In the causal view of SCL and ACL, each augmented view of the data has its own proxy label. Therefore, (A)CL aims to minimize the differences in representation output of different views of the same data. Specifically, the conditional probability $p(y^R_{ki}|x^i_k; \theta)$ and $p(y^R_{kj}|x^j_k; \theta)$ is formulated as follows:

$$p(y^R_{ki}|x^i_k; \theta) = p(\mathbf{1}_{kj}|x^i_k; \theta) = \frac{e^{\mathrm{sim}\left(f_\theta(x^i_k), f_\theta(x^j_k)\right)/t}}{\sum\limits_{x \in B^i \cup B^j \setminus \{x^i_k\}} e^{\mathrm{sim}\left(f_\theta(x^i_k), f_\theta(x)\right)/t}}, \tag{11}$$

$$p(y^R_{kj}|x^j_k; \theta) = p(\mathbf{1}_{ki}|x^j_k; \theta) = \frac{e^{\mathrm{sim}\left(f_\theta(x^j_k), f_\theta(x^i_k)\right)/t}}{\sum\limits_{x \in B^i \cup B^j \setminus \{x^j_k\}} e^{\mathrm{sim}\left(f_\theta(x^j_k), f_\theta(x)\right)/t}}, \tag{12}$$

where $\mathrm{sim}(p, q) = p^\top q / \|p\| \|q\|$ is the cosine similarity function and $t > 0$ is a temperature parameter.

Then, we have the following results:

$$
\begin{aligned}
&\arg\min_{\theta} \sum_{x_k \in U} \ell_{\mathrm{CL}}(x_k^i, x_k^j; \theta) && (13)\\
&= \arg\min_{\theta} \sum_{x_k \in U} -\log \frac{e^{\mathrm{sim}\left(f_\theta(x_k^i), f_\theta(x_k^j)\right)/t}}{\sum\limits_{x \in B^i \cup B^j \setminus \{x_k^i\}} e^{\mathrm{sim}\left(f_\theta(x_k^i), f_\theta(x)\right)/t}} - \log \frac{e^{\mathrm{sim}\left(f_\theta(x_k^j), f_\theta(x_k^i)\right)/t}}{\sum\limits_{x \in B^i \cup B^j \setminus \{x_k^j\}} e^{\mathrm{sim}\left(f_\theta(x_k^j), f_\theta(x)\right)/t}} && (14)\\
&= \arg\max_{\theta} \sum_{x_k \in U} \log \frac{e^{\mathrm{sim}\left(f_\theta(x_k^i), f_\theta(x_k^j)\right)/t}}{\sum\limits_{x \in B^i \cup B^j \setminus \{x_k^i\}} e^{\mathrm{sim}\left(f_\theta(x_k^i), f_\theta(x)\right)/t}} + \log \frac{e^{\mathrm{sim}\left(f_\theta(x_k^j), f_\theta(x_k^i)\right)/t}}{\sum\limits_{x \in B^i \cup B^j \setminus \{x_k^j\}} e^{\mathrm{sim}\left(f_\theta(x_k^j), f_\theta(x)\right)/t}} && (15)\\
&= \arg\max_{\theta} \sum_{x_k \in U} \frac{e^{\mathrm{sim}\left(f_\theta(x_k^i), f_\theta(x_k^j)\right)/t}}{\sum\limits_{x \in B^i \cup B^j \setminus \{x_k^i\}} e^{\mathrm{sim}\left(f_\theta(x_k^i), f_\theta(x)\right)/t}} + \frac{e^{\mathrm{sim}\left(f_\theta(x_k^j), f_\theta(x_k^i)\right)/t}}{\sum\limits_{x \in B^i \cup B^j \setminus \{x_k^j\}} e^{\mathrm{sim}\left(f_\theta(x_k^j), f_\theta(x)\right)/t}} && (16)\\
&= \arg\max_{\theta} \sum_{x_k \in U} p(\mathbf{1}_{kj} | x_k^i; \theta) + \sum_{x_k \in U} p(\mathbf{1}_{ki} | x_k^j; \theta) && (17)\\
&= \arg\max_{\theta} \sum_{x_k \in U} p(y_{ki}^R | x_k^i; \theta) + p(y_{kj}^R | x_k^j; \theta). && (18)
\end{aligned}
$$

Therefore, we can conclude that SCL actually maximizes the conditional probability of the proxy label given the unlabeled data, i.e., $p(y^R|x)$.

In the adversarial context, we only need to replace the natural data x_k^i and x_k^j in Eq. (13) and (18) with the adversarial data $\tilde{x}_k^i$ and $\tilde{x}_k^j$ generated in Eq. (3). Therefore, we have the following results:

$$
\begin{aligned}
&\arg\min_{\theta} \sum_{x_k \in U} \ell_{\mathrm{CL}}(x_k^i, x_k^j; \theta) + \ell_{\mathrm{CL}}(\tilde{x}_k^i, \tilde{x}_k^j; \theta)\\
&= \arg\max_{\theta} \sum_{x_k \in U} p(y_{ki}^R | x_k^i; \theta) + p(y_{kj}^R | x_k^j; \theta) + p(y_{ki}^R | \tilde{x}_k^i; \theta) + p(y_{kj}^R | \tilde{x}_k^j; \theta), \qquad (19)
\end{aligned}
$$

where the adversarial data $\tilde{x}_k^i$ and $\tilde{x}_k^j$ are generated in Eq. (3). Therefore, from the causal view, in ACL [29], maximizing both the conditional probability of the proxy label given the natural data (i.e., $p(y^R|x)$) and that given adversarial data (i.e., $p(y^R|\tilde{x})$) is equivalent to minimizing the sum of standard and adversarial contrastive losses. □

B.2 Proof of Lemma 2

Lemma 2 (restated). *AIR in Eq.* (7) *can be decomposed into two terms as follows:*

$$
\begin{aligned}
\mathcal{L}_{\mathrm{AIR}}(B; \theta, \epsilon) =& \mathbb{E}_{x \sim p^{do(\tau_i)}(\tilde{x}|x)} [\mathrm{KL}(p^{do(\tau_i)}(y^R|\tilde{x}) \| p^{do(\tau_j)}(y^R|\tilde{x}))]\\
&+ \mathbb{E}_{x \sim p^{do(\tau_i)}(y^R|\tilde{x})} [\mathrm{KL}(p^{do(\tau_i)}(\tilde{x}|x) \| p^{do(\tau_j)}(\tilde{x}|x))],
\end{aligned}
$$

where $\mathbb{E}_{x \sim Q^3(x)}[\mathrm{KL}(Q^1(x) \| Q^2(x))]$ *is the expectation of KL divergence over* Q^3 *and* Q^1, Q^2, Q^3 *are probability distributions.*

Proof. We transform AIR in Eq. (7) as follows:

$$
\begin{aligned}
\mathcal{L}_{\mathrm{AIR}}(B;\theta,\epsilon) &= \mathrm{KL}\left(p^{do(\tau_i)}(y^R|\tilde{x})p^{do(\tau_i)}(\tilde{x}|x)\|p^{do(\tau_j)}(y^R|\tilde{x})p^{do(\tau_j)}(\tilde{x}|x);B\right) && (20)\\
&= \sum_{x\in B} p^{do(\tau_i)}(y^R|\tilde{x})p^{do(\tau_i)}(\tilde{x}|x)\log\frac{p^{do(\tau_i)}(y^R|\tilde{x})p^{do(\tau_i)}(\tilde{x}|x)}{p^{do(\tau_j)}(y^R|\tilde{x})p^{do(\tau_j)}(\tilde{x}|x)} && (21)\\
&= \sum_{x\in B} p^{do(\tau_i)}(y^R|\tilde{x})p^{do(\tau_i)}(\tilde{x}|x)(\log\frac{p^{do(\tau_i)}(y^R|\tilde{x})}{p^{do(\tau_j)}(y^R|\tilde{x})}+\log\frac{p^{do(\tau_i)}(\tilde{x}|x)}{p^{do(\tau_j)}(\tilde{x}|x)}) && (22)\\
&= \sum_{x\in B} p^{do(\tau_i)}(y^R|\tilde{x})\log\frac{p^{do(\tau_i)}(y^R|\tilde{x})}{p^{do(\tau_j)}(y^R|\tilde{x})}\cdot p^{do(\tau_i)}(\tilde{x}|x)\\
&\quad + \sum_{x\in B} p^{do(\tau_i)}(\tilde{x}|x)\log\frac{p^{do(\tau_i)}(\tilde{x}|x)}{p^{do(\tau_j)}(\tilde{x}|x)}\cdot p^{do(\tau_i)}(y^R|\tilde{x}) && (23)\\
&= \sum_{x\in B}\left(\sum_{x\in\{x\}} p^{do(\tau_i)}(y^R|\tilde{x})\log\frac{p^{do(\tau_i)}(y^R|\tilde{x})}{p^{do(\tau_j)}(y^R|\tilde{x})}\right)\cdot p^{do(\tau_i)}(\tilde{x}|x)\\
&\quad + \sum_{x\in B}\left(\sum_{x\in\{x\}} p^{do(\tau_i)}(\tilde{x}|x)\log\frac{p^{do(\tau_i)}(\tilde{x}|x)}{p^{do(\tau_j)}(\tilde{x}|x)}\right)\cdot p^{do(\tau_i)}(y^R|\tilde{x}) && (24)\\
&= \sum_{x\in B}\mathrm{KL}(p^{do(\tau_i)}(y^R|\tilde{x})\|p^{do(\tau_j)}(y^R|\tilde{x};\{x\})\cdot p^{do(\tau_i)}(\tilde{x}|x)\\
&\quad + \sum_{x\in B}\mathrm{KL}(p^{do(\tau_i)}(\tilde{x}|x)\|p^{do(\tau_j)}(\tilde{x}|x);\{x\})\cdot p^{do(\tau_i)}(y^R|\tilde{x}) && (25)\\
&= \mathbb{E}_{x\sim p^{do(\tau_i)}(\tilde{x}|x)}[\mathrm{KL}(p^{do(\tau_i)}(y^R|\tilde{x})\|p^{do(\tau_j)}(y^R|\tilde{x});\{x\})]\\
&\quad + \mathbb{E}_{x\sim p^{do(\tau_i)}(y^R|\tilde{x})}[\mathrm{KL}(p^{do(\tau_i)}(\tilde{x}|x)\|p^{do(\tau_j)}(\tilde{x}|x);\{x\})] && (26)\\
&= \mathbb{E}_{x\sim p^{do(\tau_i)}(\tilde{x}|x)}[\mathrm{KL}(p^{do(\tau_i)}(y^R|\tilde{x})\|p^{do(\tau_j)}(y^R|\tilde{x}))]\\
&\quad + \mathbb{E}_{x\sim p^{do(\tau_i)}(y^R|\tilde{x})}[\mathrm{KL}(p^{do(\tau_i)}(\tilde{x}|x)\|p^{do(\tau_j)}(\tilde{x}|x))] && (27)
\end{aligned}
$$

Eq. (21) and (25) are obtained by using the KL divergence $\mathrm{KL}(p(x)\|q(x);B)=\sum_{x\in B}p(x)\log\frac{p(x)}{q(x)}$. We obtained Eq. (27) by omiting the term $\{x\}$ in Eq. (26) for simplicity and finished proving Lemma 2. □

B.3 Proof of Proposition 3

Proposition 3 (restated). *AIR implicitly enforces the robust representation to satisfy the following two proxy criteria:*

$$
(1)\quad p^{do(\tau_i)}(y^R|\tilde{x})=p^{do(\tau_j)}(y^R|\tilde{x}),\qquad (2)\quad p^{do(\tau_i)}(\tilde{x}|x)=p^{do(\tau_j)}(\tilde{x}|x).
$$

Proof. To enforce the robust representation to satisfy the two proxy criteria shown in Eq. (28), we need to regulate the ACL with the following regularization

$$
\begin{aligned}
&\mathrm{KL}\left(p^{do(\tau_i)}(y^R|\tilde{x})\|p^{do(\tau_j)}(y^R|\tilde{x});B\right)+\mathrm{KL}\left(p^{do(\tau_i)}(\tilde{x}|x)\|p^{do(\tau_j)}(\tilde{x}|x);B\right) && (28)\\
&= \sum_{x\in B} p^{do(\tau_i)}(y^R|\tilde{x})\log\frac{p^{do(\tau_i)}(y^R|\tilde{x})}{p^{do(\tau_j)}(y^R|\tilde{x})}+\sum_{x\in B} p^{do(\tau_i)}(\tilde{x}|x)\log\frac{p^{do(\tau_i)}(\tilde{x}|x)}{p^{do(\tau_j)}(\tilde{x}|x)} && (29)\\
&= \sum_{x\in B}\mathrm{KL}(p^{do(\tau_i)}(y^R|\tilde{x})\|p^{do(\tau_j)}(y^R|\tilde{x};\{x\})+\sum_{x\in B}\mathrm{KL}(p^{do(\tau_i)}(\tilde{x}|x)\|p^{do(\tau_j)}(\tilde{x}|x);\{x\}). && (30)
\end{aligned}
$$

Compared with Eq. (25) in the proof of Lemma 2, we can find that AIR also penalizes the same two KL divergence terms while AIR has two extra calibration terms $p^{do(\tau_i)}(\tilde{x}|x)$ and $p^{do(\tau_i)}(y^R|\tilde{x})$ that adjust the confidence of each KL divergence term, respectively. We empirically studied the impact of the calibration term in Appendix C.8 and found that the calibration term is beneficial to improving the performance of ACL. Therefore, it indicates that AIR implicitly enforces the robust representation to satisfy the two proxy criteria shown in Eq. (28). □

B.4 Proof of Proposition 4

Proposition 4 (restated). *Let $\mathcal{Y} = \{y_t\}_{t=1}^T$ be a label set of a downstream classification task, $\mathcal{Y}^R$ be a refinement of $\mathcal{Y}$, and $\tilde{x}_t$ be the adversarial data generated on the downstream task. Assuming that $\tilde{x}_t \in \mathcal{B}_\epsilon[x]$ and $\tilde{x} \in \mathcal{B}_\epsilon[x]$, we have the following results:*

$$p^{do(\tau_i)}(y^R|\tilde{x}) = p^{do(\tau_j)}(y^R|\tilde{x}) \Longrightarrow p^{do(\tau_i)}(y_t|\tilde{x}_t) = p^{do(\tau_j)}(y_t|\tilde{x}_t) \quad \forall \tau_i, \tau_j \in \mathcal{T},$$
$$p^{do(\tau_i)}(\tilde{x}|x) = p^{do(\tau_j)}(\tilde{x}|x) \Longrightarrow p^{do(\tau_i)}(\tilde{x}_t|x) = p^{do(\tau_j)}(\tilde{x}_t|x) \quad \forall \tau_i, \tau_j \in \mathcal{T}.$$

Proof. The proof is inspired by Mitrovic et al. [38].

For $t \in \{1, \ldots, T\}$, we have

$$p^{do(\tau_i)}(y_t|\tilde{x}_t) = \int p^{do(\tau_i)}(y_t|y^R) p^{do(\tau_i)}(y^R|\tilde{x}_t) dy^R \tag{31}$$
$$= \int p(y_t|y^R) p^{do(\tau_i)}(y^R|\tilde{x}_t) dy^R \tag{32}$$
$$= \int p(y_t|y^R) p^{do(\tau_j)}(y^R|\tilde{x}_t) dy^R \tag{33}$$
$$= p^{do(\tau_j)}(y_t|\tilde{x}_t). \tag{34}$$

Eq. (31) holds due to the fact that y^R is a refinement of y_t according to the definition of the refinement shown in Mitrovic et al. [38]. Eq. (32) holds since the relationship between y_t and y^R is independent of the style, i.e., $p^{do(\tau_i)}(y_t|y^R) = p^{do(\tau_j)}(y_t|y^R)$. Eq. (33) holds because of the assumption that the prediction of adversarial data is independent of the style. Due to that the condition $p^{do(\tau_i)}(y^R|\tilde{x}) = p^{do(\tau_j)}(y^R|\tilde{x})$ holds for any $\tilde{x} \in \mathcal{B}_\epsilon[x]$, thereby, this condition also holds for $\tilde{x}_t \in \mathcal{B}_\epsilon[x]$. Lastly, we obtain that the prediction of adversarial data will be still independent of the style factors on downstream tasks.

Next, due to that the condition $p^{do(\tau_i)}(\tilde{x}|x) = p^{do(\tau_j)}(\tilde{x}|x)$ holds for any $\tilde{x} \in \mathcal{B}_\epsilon[x]$, this condition will hold for $\tilde{x}_t \in \mathcal{B}_\epsilon[x]$ as well, i.e., $p^{do(\tau_i)}(\tilde{x}_t|x) = p^{do(\tau_j)}(\tilde{x}_t|x)$. Therefore, the consistency between adversarial and natural data will be still invariant of the style factors on the downstream tasks. □

C Extensive Experimental Details and Results

Experimental environments. We conducted all experiments on Python 3.8.8 (PyTorch 1.13) with NVIDIA RTX A5000 GPUs (CUDA 11.6).

Pre-training details of ACL [29]. Following Jiang et al. [29], we leveraged ResNet-18 [25] with the dual batch normalization (BN) [46] as the representation extractor, where one BN is used for the standard branch of the feature extractor and the other BN is used for the adversarial branch, during conducting ACL [29] and its variant DynACL [36]. We pre-trained ResNet-18 models using SGD for 1000 epochs with an initial learning rate 5.0 and a cosine annealing schedule [35]. During pre-training, we set the adversarial budget ϵ as $8/255$, the hyperparameter ω as 0.0, and the strength of data augmentation as 1.0. As for the reproduction of the baseline, we used the pre-trained weights published in the GitHub of ACL[4][5] as the pre-trained representation extractor for finetuning.

Pre-training details of DynACL [36]. The training configurations of DynACL [36] followed ACL [29], except for the strength of data augmentation and the hyperparameter ω. We denote the

[4] Link of pre-trained weights via ACL on CIFAR-10.

[5] Link of pre-trained weights via ACL on CIFAR-100.

https://doi.org/10.52202/075280-0734

Table 9: The robust/standard test accuracy (%) achieved by ACL-AIR with different λ_1 and λ_2.

λ_2 λ_1	0.00	0.25	0.50	1.00
0.00	37.39/78.27	38.61/79.48	38.70/79.96	38.76/79.81
0.25	37.55/78.53	38.70/79.53	38.79/79.65	38.73/79.76
0.50	37.51/78.97	38.63/79.72	**38.89/80.03**	38.82/80.08
1.00	37.04/79.17	38.24/79.64	38.39/80.09	38.77/79.83

Table 10: Performance in semi-supervised settings evaluated on the CIFAR-10 task.

Label	ACL [29]		ACL-AIR		DynACL [36]		DynACL-AIR	
ratio	AA (%)	SA (%)	AA (%)	SA (%)	AA (%)	SA (%)	AA (%)	SA (%)
1%	45.63	74.84	**45.97**	**76.63**	45.78	76.89	**46.25**	**78.57**
10%	45.68	76.30	**46.17**	**77.52**	47.08	78.22	**47.41**	**79.94**

strength of data augmentation and the hyperparameter at epoch e as μ_e and ω_e respectively, where

$$\mu_e = 1 - \lfloor \frac{e}{K} \rfloor \cdot \frac{K}{E}, \quad e \in \{0, 1, \ldots, E-1\} \tag{35}$$

$$\omega_e = \nu \cdot (1 - \mu_e), \quad e \in \{0, 1, \ldots, E-1\} \tag{36}$$

in which the decay period $K = 50$, the reweighting rate $\nu = 2/3$, the total training epoch $E = 1000$. In our implementation of DynACL, we only take the dynamic strategy and do not take the trick of the stop gradient operation and the momentum encode [24, 11]. As for the reproduction of the baseline, we downloaded the pre-trained weights published in the GitHub of DynACL[6][7][8] as the pre-trained representation extractor for finetuning.

Details of finetuning procedures. As for SLF and ALF, we fixed the parameters of the representation extractor and only finetuned the linear classifier using the natural training data and adversarial training data respectively. As for AFF, we finetuned all the parameters using the adversarial training data. For all finetuning procedures, we used SGD for 25 epochs for linear finetuning and full finetuning respectively and momentum is set as $2e-4$. The initial learning of linear finetuning is set as 0.01 on CIAFR-10, 0.05 on CIFAR-100, and 0.1 on STL-10. The adversarial budget is fixed as $8/255$ for ALF and AFF. In practice, we used the finetuning code published in the GitHub of DynACL for implementing finetuning procedures.

C.1 Performance in Semi-Supervised Settings [1]

Table 10 reports the performance evaluated in semi-supervised settings [1] during the finetuning procedure. In semi-supervised settings, following ACL [29] and DynACL [36], we first standardly finetuned the pre-trained model using the labelled data and then generated the pseudo labels using the standardly finetuned model. Then, we finetuned the model using the data with pseudo labels as well as the labelled data via AFF. The results validate that AIR ($\lambda_1 = 0.5, \lambda_2 = 0.5$) can consistently enhance both robust and standard test accuracy of ACL and its variant DynACL in semi-supervised settings.

C.2 Impact of the Hyper-Parameters λ_1 and λ_2

We show the performance achieved by ACL-AIR of different $\lambda_1 \in \{0.00, 0.25, 0.50, 1.00\}$ and $\lambda_2 \in \{0.00, 0.25, 0.50, 1.00\}$ on the CIFAR-10 task in Table 9. We can notice that when only leveraging adversarial regularization ($\lambda_1 = 0, \lambda_2 > 0$), both standard and robust test accuracy gain significant improvement, which indicates that adversarial regularization serves an important role in improving the standard generalization and robustness of ACL. By incorporating adversarial and standard regularization together (when both $\lambda_1 > 0$ and $\lambda_2 > 0$), the standard generalization of ACL gets further improved. Table 9 guides us to set $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$ since it can yield the best performance.

C.3 Incorporating AIR with Two Extra Variants of ACL

Here, we show extra results to validate that our proposed AIR can enhance two extra variants of ACL, including AdvCL [22] and AdvCL with A-InfoNCE [50].

[6] Link of pre-trained weights via DynACL on CIFAR-10.

[7] Link of pre-trained weights via DynACL on CIFAR-100.

[8] Link of pre-trained weights via DynACL on STL-10.

Table 11: Self-task adversarial robustness transferability evaluated on CIFAR-10 using various backbone networks including ResNet-34, ResNet-50, and WRN-28-10.

Network	Pre-training	AA (%)	SA (%)
ResNet-34	DynACL [36]	47.01	78.84
	DynACL-AIR	**47.56**	**80.67**
ResNet-50	DynACL [36]	47.19	79.65
	DynACL-AIR	**47.82**	**81.27**
WRN-28-10	DynACL [36]	41.13	71.23
	DynACL-AIR	**43.86**	**73.41**

Table 12: Self-task adversarial robustness transferability evaluated on CIFAR-10 achieved by AdvCL [22], AdvCL-AIR, A-InfoNCE [50], and A-InfoNCE-AIR.

Pre-training	AdvCL [22]	AdvCL-AIR	A-InfoNCE [50]	A-InfoNCE-AIR
AA (%)	42.58	**43.24**	42.68	**42.84**
SA (%)	80.78	**81.53**	83.18	**83.99**

Table 13: Self-task adversarial robustness transferability evaluated on the CIFAR-10 dataset via SLF under various attacks including APGD-CE [13], APGD-DLR [13], FAB [14], and Square Attack [2].

Pre-training	APGD-CE (%)	APGD-DLR (%)	FAB (%)	Square Attack (%)
ACL [29]	39.66	41.18	39.27	48.98
ACL-AIR	**40.39**	**40.98**	**41.01**	**49.80**
DynACL [36]	46.28	46.41	45.56	50.06
DynACL-AIR	**47.05**	**47.29**	**46.02**	**50.58**

AdvCL [22] leverages an extra contrastive view of high-frequency components and the pseudo labels generated by the clustering method. We leveraged our proposed method IR ($\lambda_1 = 0.5, \lambda_2 = 0.5$) to further improve the performance of AdvCL. Yu et al. [50] proposed an asymmetric InfoNCE objective (A-InfoNCE) that treats adversaries as inferior positives that induce weaker learning signals, or as hard negatives exhibiting higher contrast to other negative samples.

We used the code provide by Yu et al. [50] in their GitHub[9] to implement the pre-training of AdvCL as well as A-InfoNCE, and the finetuning procedure. In Table 12, we show that our proposed AIR ($\lambda_1 = 0.5, \lambda_2 = 0.5$) can consistently improve the robust and standard test accuracy of two extra variants [22, 50].

C.4 Applicability with Different Backbone Networks

In this subsection, we demonstrate that DynACL-AIR can consistently improve the performance on ResNet-34, ResNet-50 [25], and WRN-28-10 [51]. We followed the same training configurations of pre-training and finetuning (SLF) in Section 4 except for the backbone network. Table 11 validates that IR can further enhance the robust and standard test accuracy of DynACL on various backbone networks.

We observe that WRN-28-10 yields a worse performance compared to ResNet-50. We conjecture that it is because we set the batch size to 128 during pre-training due to the limitation of our GPU memory. We believe using a larger batch size during pre-training can further improve the performance of WRN-28-10 according to Chen et al. [9].

C.5 Robustness Evaluation under Various Attacks

In this subsection, we conducted the robustness self-transferability against three strong white-box attacks (APGD-CE [13], APGD-DLR [13] and FAB [14]) and one strong black-box attack (i.e., Square Attack [2]). We evaluate the robustness on the CIFAR-10 dataset via SLF and report the results in Table 13. The results validate that our proposed method can consistently improve robust test accuracy over various adversaries.

C.6 Test Accuracy under Each Type of Common Corruption

We report the test accuracy under each type of common corruption [26] with corruption severity being fixed as 5 in Table 14. We used pre-trained models on CIFAR-10 via DynACL and DynACL-AIR.

[9] GitHub provided by Yu et al. [50].

https://doi.org/10.52202/075280-0734

Table 14: Test accuracy evaluated on CIFAR-10-C under each type of common corruptions with corruption severity being fixed as 5 of CIFAR-10 pre-trained models after SLF and AFF.

Pre-training	Finetuning	Noise			Blur				Weather				Digital			
		Gaussian	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
DynACL [36]	SLF	68.14	68.22	62.76	69.73	68.84	67.01	71.86	64.73	62.36	26.24	66.53	17.66	70.64	72.44	73.63
DynACL-AIR		**70.10**	**70.47**	**64.40**	**69.79**	**71.18**	**67.90**	**72.10**	**66.72**	**63.94**	**30.16**	**67.27**	**18.21**	**72.43**	**75.17**	**76.80**
DynACL [36]	AFF	74.02	74.62	66.87	71.44	73.21	69.22	73.68	67.83	66.29	27.59	69.27	18.72	74.60	77.50	79.05
DynACL-AIR		**74.54**	**75.13**	**67.06**	**72.63**	**74.78**	**70.39**	**75.66**	**69.07**	**67.47**	**28.74**	**71.24**	**19.02**	**75.85**	**79.09**	**80.64**

Table 15: Standard deviation of robustness self-transferability against common corruptions reported in Table 2.

Dataset	Pre-training	SLF			ALF			AFF		
		CS-1	CS-3	CS-5	CS-1	CS-3	CS-5	CS-1	CS-3	CS-5
CIFAR-10	ACL [29]	0.06	0.25	0.13	0.13	0.17	0.10	0.11	0.17	0.21
	ACL-AIR	0.03	0.02	0.02	0.07	0.09	0.08	0.16	0.20	0.14
	DynACL [36]	0.02	0.01	0.02	0.07	0.16	0.09	0.13	0.20	0.18
	DynACL-AIR	0.01	0.03	0.03	0.07	0.11	0.14	0.16	0.11	0.10
CIFAR-100	ACL [29]	0.01	0.09	0.02	0.12	0.14	0.11	0.09	0.04	0.17
	ACL-AIR	0.04	0.02	0.02	0.13	0.09	0.04	0.17	0.04	0.16
	DynACL [36]	0.03	0.02	0.01	0.14	0.07	0.12	0.08	0.09	0.16
	DynACL-AIR	0.08	0.05	0.02	0.08	0.12	0.05	0.16	0.16	0.10

Table 16: Standard deviation of robustness self-transferability against common corruptions reported in Table 4.

$\mathcal{D}_1 \to \mathcal{D}_2$	Pre-training	SLF			ALF			AFF		
		CS-1	CS-3	CS-5	CS-1	CS-3	CS-5	CS-1	CS-3	CS-5
CIFAR-10 $\to$ CIFAR-100	ACL [29]	0.05	0.02	0.03	0.07	0.09	0.05	0.15	0.10	0.31
	ACL-AIR	0.01	0.04	0.04	0.09	0.12	0.10	0.11	0.16	0.15
	DynACL [36]	0.04	0.04	0.05	0.11	0.07	0.07	0.19	0.18	0.10
	DynACL-AIR	0.03	0.02	0.03	0.12	0.13	0.09	0.18	0.04	0.07
CIFAR-100 $\to$ CIFAR-10	ACL [29]	0.03	0.03	0.03	0.08	0.09	0.14	0.08	0.07	0.15
	ACL-AIR	0.03	0.02	0.03	0.09	0.15	0.13	0.13	0.14	0.20
	DynACL [36]	0.03	0.01	0.02	0.11	0.12	0.14	0.05	0.14	0.19
	DynACL-AIR	0.02	0.08	0.02	0.06	0.08	0.09	0.14	0.21	0.17

Table 17: Impact of calibration terms evaluated on the CIFAR-10 task.

Pre-training	Calibration	SLF		ALF		AFF	
		AA (%)	SA (%)	AA (%)	SA (%)	AA (%)	SA (%)
ACL-AIR	×	38.55	79.80	40.80	77.57	49.51	81.95
ACL-AIR	√	**38.70**	**79.96**	**41.09**	**77.99**	**49.59**	**82.30**
DynACL-AIR	×	45.09	77.79	46.01	76.12	50.54	82.35
DynACL-AIR	√	**45.23**	**78.01**	**46.12**	**77.01**	**50.66**	**82.62**

Experimental settings are the same as Section 4. The results validate that IR can consistently improve the test accuracy under each type of common corruption.

C.7 Standard Deviation of Test Accuracy under Common Corruptions

Here, we provide the standard deviation of the test accuracy under common corruptions in Tables 15 and 16.

C.8 Impact of Calibration Terms

In the proof of Proposition 3, we show that there exist calibration terms $p^{do(\tau_i)}(\tilde{x}|x)$ and $p^{do(\tau_i)}(y^R|\tilde{x})$ in AIR. Here, we empirically investigate the impact of calibration on the performance in Table 17. The results empirically validate that the calibration term is an essential component of AIR to adjust the confidence of the KL divergence and further enhance the performance of ACL.

D Limitations

One of the limitations is that similar to previous works [29, 22, 50, 36], our proposed AIR is not helpful in improving the efficiency and scalability of ACL methods. Therefore, robust pre-training via

https://doi.org/10.52202/075280-0734

ACL-AIR is still unfriendly to the environment due to emitting much carbon dioxide and consuming much electricity. Besides, applying ACL-AIR to large-scale datasets such as ImageNet-21K [41] is still computationally prohibitive with limited GPU sources.

E Possible Negative Societal Impacts

Our proposed method aims to improve the performance of robust self-supervised pre-training methods [29, 22, 50, 36]. The robust self-supervised pre-training procedure is extremely time-consuming since it needs to spend much time generating adversarial training data, which thus leads to consuming a lot of electricity and emitting lots of carbon dioxide. Unfortunately, our proposed method is not helpful in improving the efficiency of ACL. Consequently, our study could exacerbate the greenhouse effect and be not conducive to environmental protection.

https://doi.org/10.52202/075280-0734