Towards Robust Foundation Models

Efficient and Effective Adversarial Contrastive Learning

Xilie Xu

Ph.D. student, School of Computing, NUS

28th Nov 2023

Advisor: Prof. Mohan Kankanhalli

Towards Robust Foundation Models

Efficient and Effective Adversarial Contrastive Learning

Foundation Model

Large-scale unlabeled datasets

Self-supervised learning
(e.g., Standard contrastive learning)

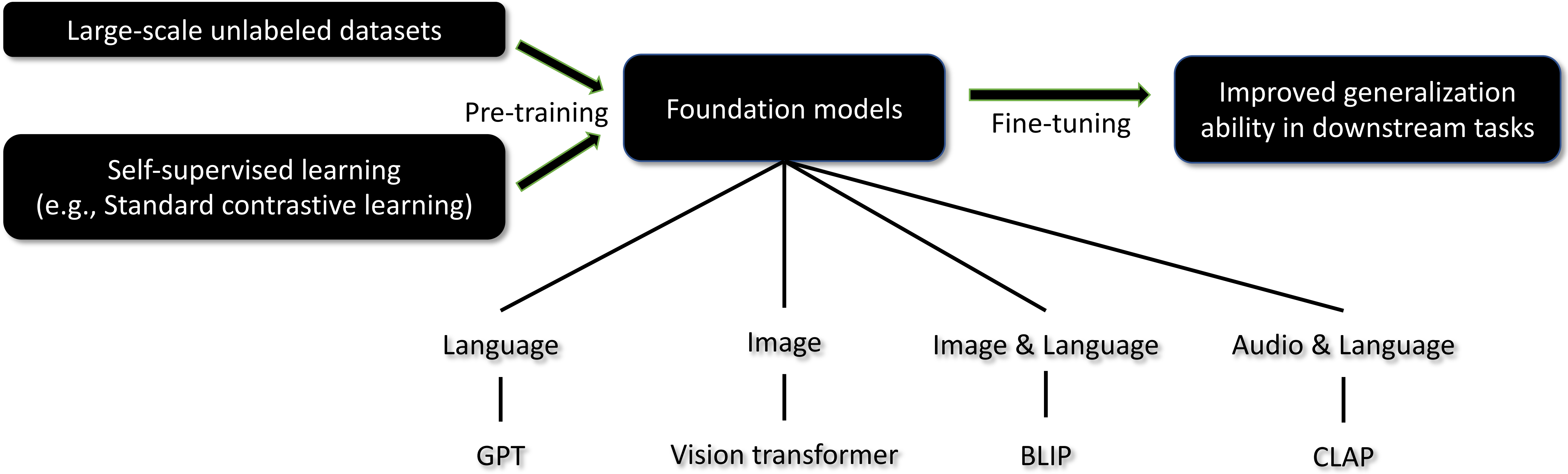
Pre-training

Foundation models

Fine-tuning

Improved generalization
ability in downstream tasks

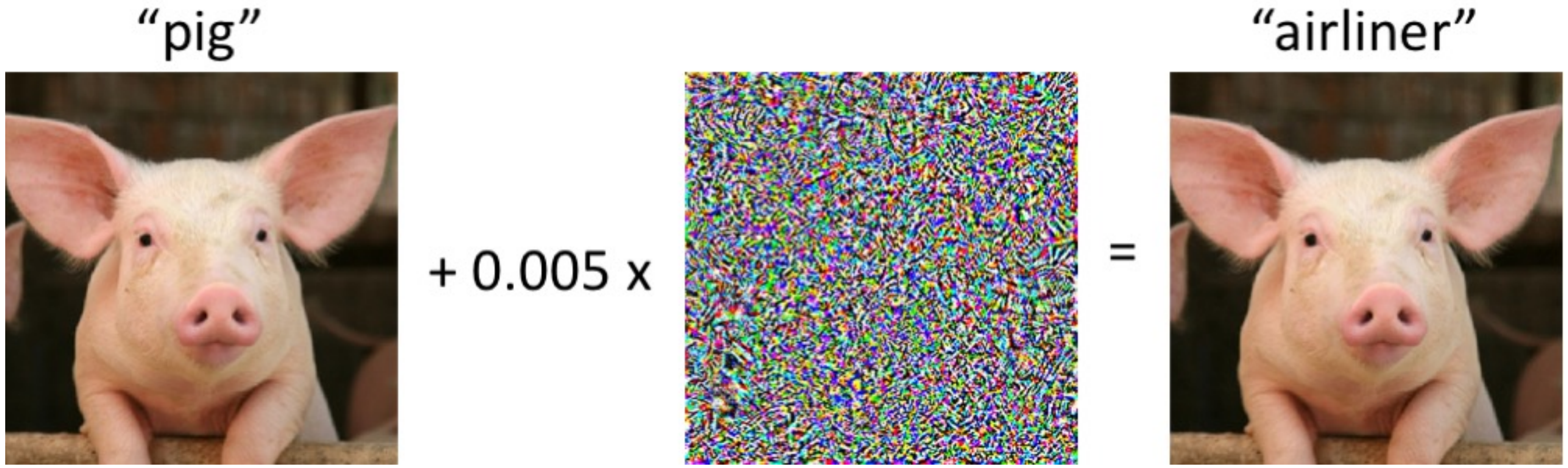
Foundation Model



Towards Robust Foundation Models

Efficient and Effective Adversarial Contrastive Learning

Security Risk---Adversarial Attack



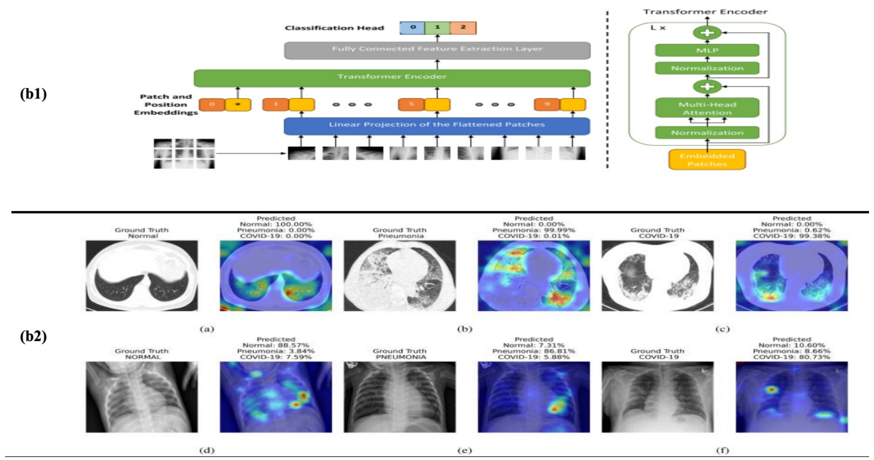
Natural data x + Imperceptible adversarial perturbation = Adversarial data $\tilde{x}$

Image from https://gradientscience.org/intro_adversarial/

Security Risk---Adversarial Attack

Potential security risks when applying foundation models to safety-critical tasks

Medical diagnosis



[Henrya et al., ArXiv 2022]

Traffic sign recognition

Image from https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.itm-p.com%2Fprotect-iot-applications-from-adversarial-evasion-attacks&psig=AOvVaw2ltdTWXQ51FptX2EFu9gKr&ust=1700842535801000&source=images&cd=vfe&opi=89978449&ved=0CBQQjhxqFwoTCJidm6vC2oIDFQAAAAAdAAAAABAJ

Risks Urge Robust Foundation Models

Robust foundation models should:

1. be generalizable to downstream tasks;
2. be robust against adversarial attacks.

Robust foundation models $\xrightarrow{\text{Fine-tuning}}$ Improved generalization ability and adversarial robustness in downstream tasks

Risks Urge Robust Foundation Models

Robust foundation models should:

1. be generalizable to downstream tasks;
2. be robust against adversarial attacks.

(Large-scale) unlabeled datasets

Robust self-supervised learning (e.g., Adversarial Contrastive Learning)

Pre-training

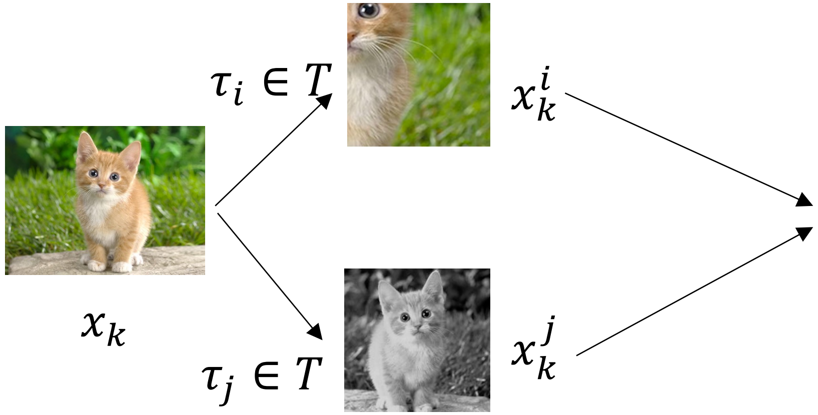
Robust foundation models

Fine-tuning

Improved generalization ability and adversarial robustness in downstream tasks

Adversarial Contrastive Learning (ACL)

Data augmentations
τ_i and τ_j



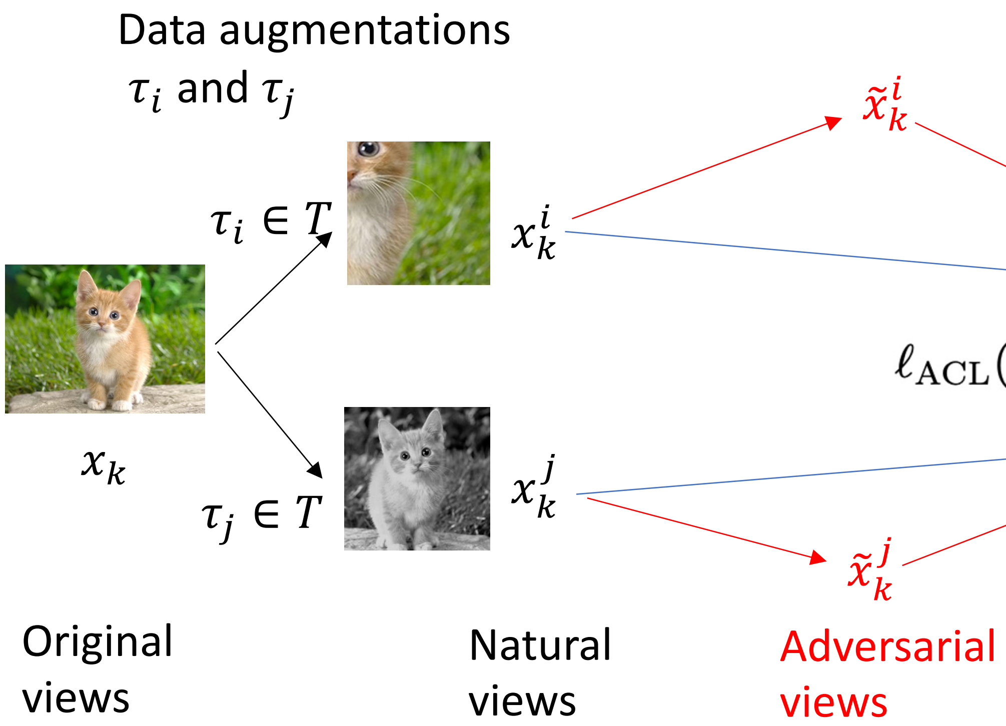
Standard Contrastive Learning

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

Original views

Natural views

Adversarial Contrastive Learning (ACL)



Adversarial Contrastive Learning

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

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

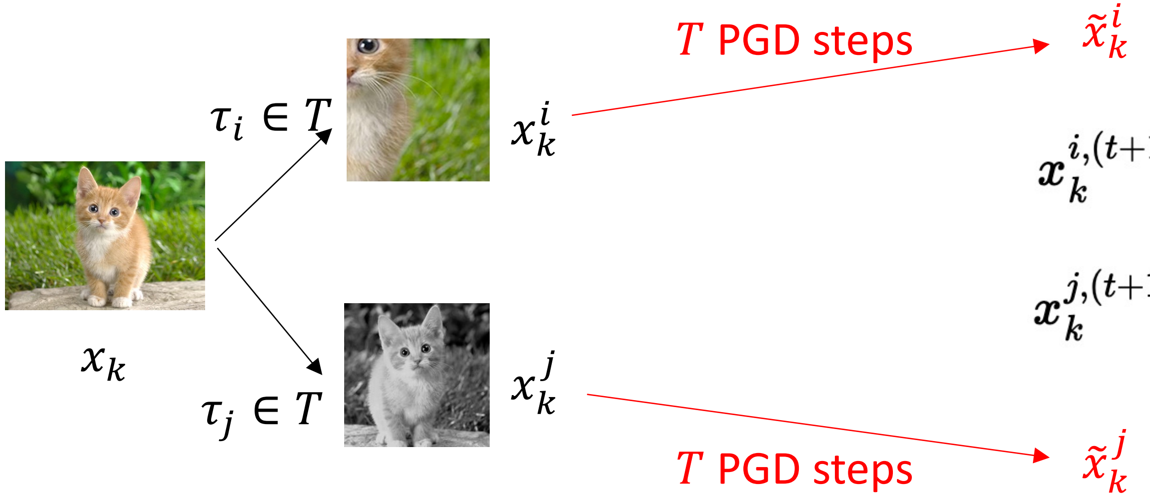
At each epoch, the ACL alternatively conducts Step (1) and (2):

Step (1): Generating adversarial views
Step (2): Updating parameters via minimizing the contrastive loss on the natural views and adversarial views.

Towards Robust Foundation Models

Efficient and Effective Adversarial Contrastive Learning

Motivation: ACL is inefficient due to T PGD steps



$$x_k^{i,(t+1)} = \Pi_{\mathcal{B}_\epsilon[x_k^{i,(0)}]}\left(x_k^{i,(t)} + \rho \cdot \mathrm{sign}(\nabla_{x_k^{i,(t)}} \ell_{\mathrm{CL}}(x_k^{i,(t)}, x_k^{j,(t)}))\right)$$

$$x_k^{j,(t+1)} = \Pi_{\mathcal{B}_\epsilon[x_k^{j,(0)}]}\left(x_k^{j,(t)} + \rho \cdot \mathrm{sign}(\nabla_{x_k^{j,(t)}} \ell_{\mathrm{CL}}(x_k^{i,(t)}, x_k^{j,(t)}))\right)$$

ACL on the entire training set is extremely time-consuming.

- CIFAR-10: about *43 hours*
- ImageNet-1K: about *650 hours*

Project Gradient Descent (PGD)

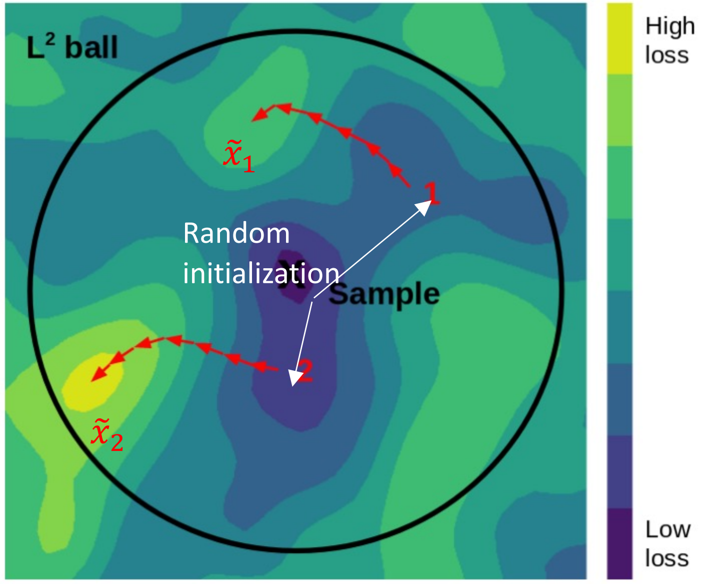
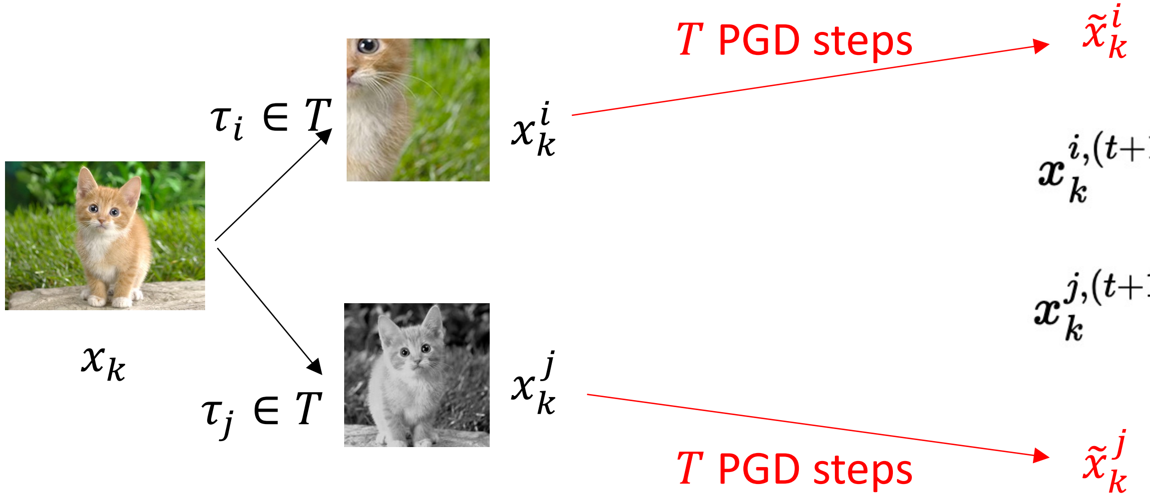


Image modified from https://towardsdatascience.com/know-your-enemy-7f7c5038bdf3

Motivation: ACL is inefficient due to T PGD steps



$$x_k^{i,(t+1)} = \Pi_{\mathcal{B}_\epsilon[x_k^{i,(0)}]}\left(x_k^{i,(t)} + \rho \cdot \mathrm{sign}(\nabla_{x_k^{i,(t)}} \ell_{\mathrm{CL}}(x_k^{i,(t)}, x_k^{j,(t)})\right)$$

$$x_k^{j,(t+1)} = \Pi_{\mathcal{B}_\epsilon[x_k^{j,(0)}]}\left(x_k^{j,(t)} + \rho \cdot \mathrm{sign}(\nabla_{x_k^{j,(t)}} \ell_{\mathrm{CL}}(x_k^{i,(t)}, x_k^{j,(t)})\right)$$

ACL on the entire training set is extremely time-consuming.

- CIFAR-10: about *43 hours*
- ImageNet-1K: about *650 hours*

How can we speed up ACL?

Project Gradient Descent (PGD)

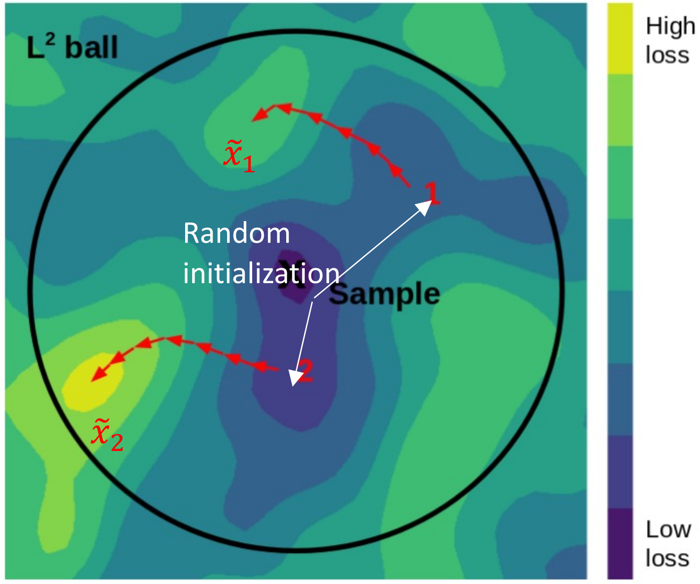


Image modified from https://towardsdatascience.com/know-your-enemy-7f7c5038bdf3

Robustness-Aware Coreset Selection (RCS)

- Intuitive idea: Find an informative training subset (called "coreset")
  - Decreasing the number of training samples
  - Guaranteeing the model to effectively learn robust representations

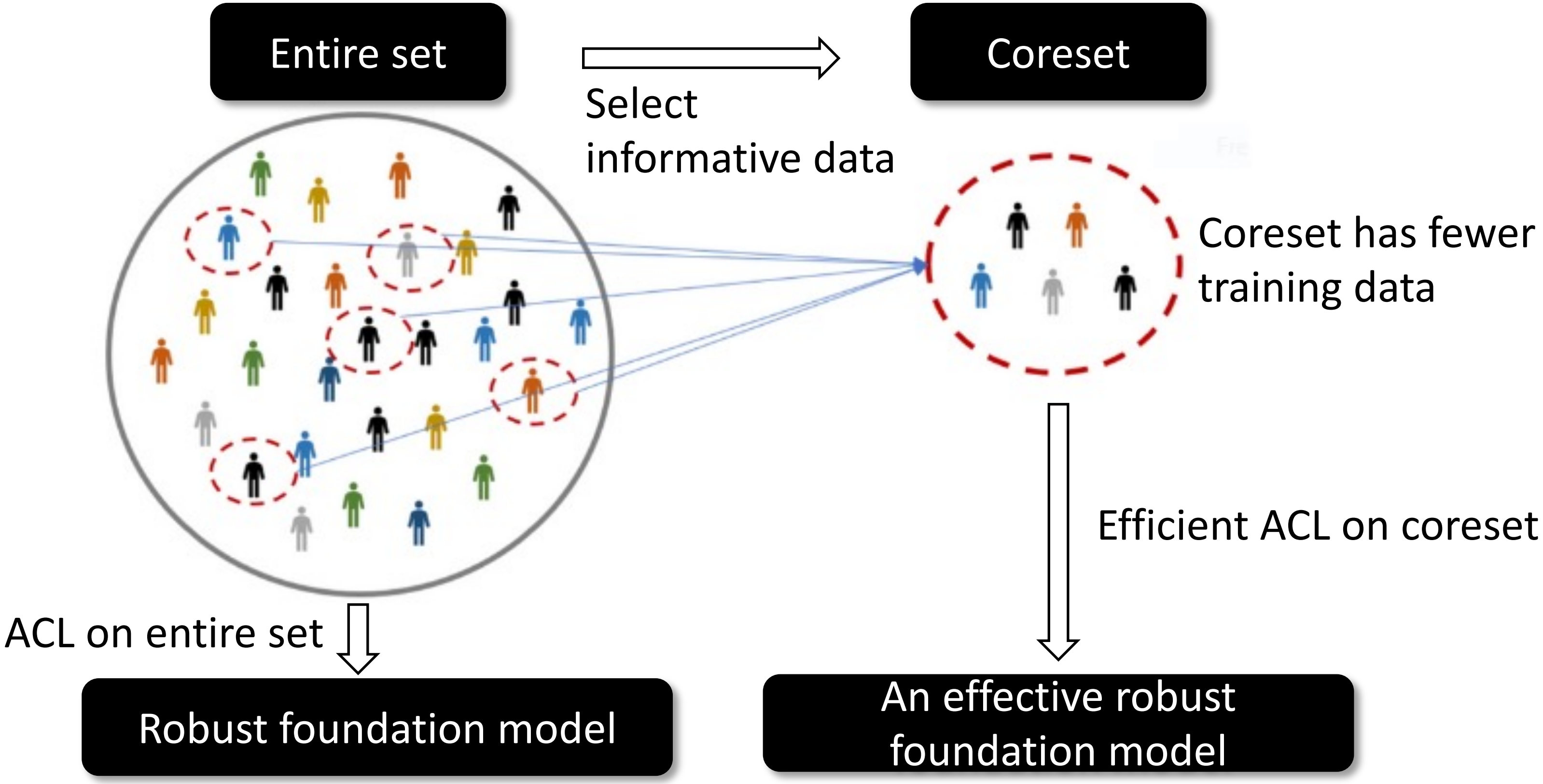


Image from https://medium.com/analytics-vidhya/sampling-statistical-approach-in-machine-learning-4903c40ebf86

Robustness-Aware Coreset Selection (RCS)

- Intuitive idea: Find an informative training subset (called "coreset")
- Objective function of RCS

ACL on CIFAR-10		SLF on CIFAR-10	SLF on CIFAR-10
	RD loss (lower is better)	SA (%)	RA (%)
Entire	0.1243	78.87	39.19
Random-0.05	0.3357	67.45	22.96

Representational divergence (RD)

$$\ell_{\mathrm{RD}}(x;\theta) = d(g \circ f_\theta(\tilde{x}), g \circ f_\theta(x)) \quad \text{s.t.} \quad \tilde{x} = \arg\max_{x' \in \mathcal{B}_\epsilon[x]} d(g \circ f_\theta(x'), g \circ f_\theta(x))$$

RD, without using labels, measures the adversarial robustness.

Robustness-Aware Coreset Selection (RCS)

- Intuitive idea: Find an informative training subset (called "coreset")
- Objective function of RCS

Coreset

Subset fraction

Unlabeled validation set

$$S^* = \mathop{\arg\min}_{S \subseteq X, |S|/|X| = k} \mathcal{L}_{\mathrm{RD}}(U; \theta(S))$$

Representational divergence (RD)

$$\theta(S) = \mathop{\arg\min}_{\theta} \mathcal{L}_{\mathrm{ACL}}(S; \theta),$$

Adversarial contrastive loss

$$\ell_{\mathrm{RD}}(x;\theta) = d(g \circ f_\theta(\tilde{x}), g \circ f_\theta(x)) \quad \text{s.t.} \quad \tilde{x} = \mathop{\arg\max}_{x' \in \mathcal{B}_\epsilon[x]} d(g \circ f_\theta(x'), g \circ f_\theta(x))$$

RD, without using labels, measures the adversarial robustness.

Training on coreset

$$\theta(S) = \mathop{\arg\min}_{\theta} \mathcal{L}_{\mathrm{ACL}}(S; \theta)$$

$\Downarrow$

Achieve the minimized RD

$$\mathop{\arg\min}_{S \subseteq X, |S|/|X| = k} \mathcal{L}_{\mathrm{RD}}(U; \theta(S))$$

$\Downarrow$

Effectively obtain adversarial robustness

RCS is a problem of set function maximization

$$\theta(S) = \arg\min_{\theta} \mathcal{L}_{\mathrm{ACL}}(S;\theta)$$

$$S^* = \arg\min_{S \subseteq X, |S|/|X|=k} \mathcal{L}_{\mathrm{RD}}(U;\theta(S)) \longrightarrow G_\theta(S \subseteq X) \triangleq -\mathcal{L}_{\mathrm{RD}}(U;\theta(S)) = -\mathcal{L}_{\mathrm{RD}}(U;\theta - \eta\nabla_\theta \mathcal{L}_{\mathrm{ACL}}(S;\theta))$$

Set function

One-step gradient approximation

$$S^* = \arg\max_{S \subseteq X, |S|/|X|=k} G_\theta(S)$$

Transform into a problem of maximizing a set function subject to a constraint on the size of the set

RCS via greedy search

RCS greedily finds and adds the data which has the largest marginal grain into the coreset.

$$
\begin{aligned}
G_\theta(x \mid S) &= G_\theta(S \cup \{x\}) - G_\theta(S) \\
&= -\mathcal{L}_{\mathrm{RD}}(U; \theta - \eta \nabla_\theta \mathcal{L}_{\mathrm{ACL}}(S;\theta) - \eta \nabla_\theta \mathcal{L}_{\mathrm{ACL}}(\{x\};\theta)) \\
&\quad + \mathcal{L}_{\mathrm{RD}}(U; \theta - \eta \nabla_\theta \mathcal{L}_{\mathrm{ACL}}(S;\theta)) \\
&\approx -(\mathcal{L}_{\mathrm{RD}}(U; \theta - \eta \nabla_\theta \mathcal{L}_{\mathrm{ACL}}(S;\theta)) - \eta \nabla_\theta \mathcal{L}_{\mathrm{RD}}(U;\theta_S)^\top \nabla_\theta \mathcal{L}_{\mathrm{ACL}}(\{x\};\theta) + \xi) \\
&\quad + \mathcal{L}_{\mathrm{RD}}(U; \theta - \eta \nabla_\theta \mathcal{L}_{\mathrm{ACL}}(S;\theta)) \\
&\approx \eta \boxed{\nabla_\theta \mathcal{L}_{\mathcal{RD}}(U; \theta - \eta \nabla_\theta \mathcal{L}_{\mathrm{ACL}}(S;\theta))^\top} \boxed{\nabla_\theta \mathcal{L}_{\mathrm{ACL}}(\{x\};\theta)}
\end{aligned}
$$

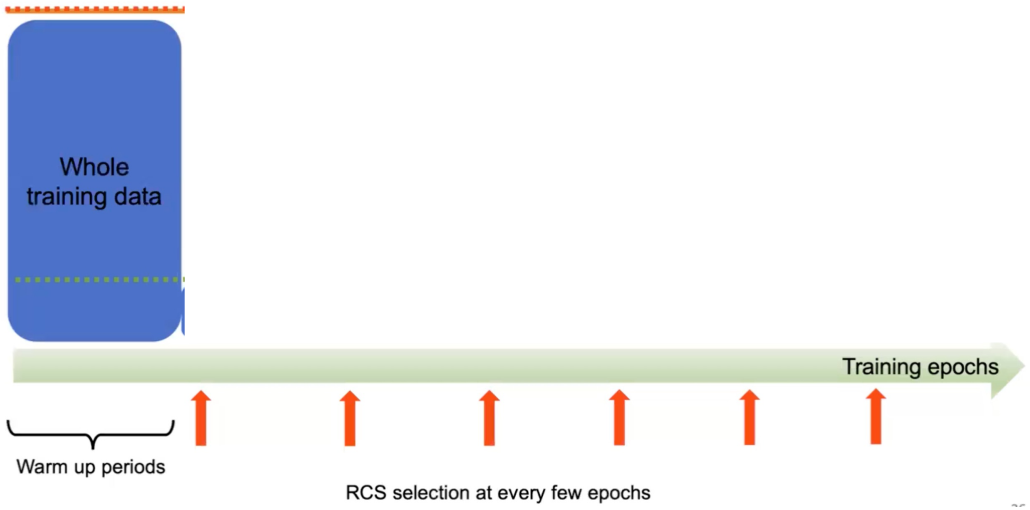
Derived via Taylor expansion

Marginal gain = Similarity (validation loss gradient, training loss gradient)

More adversarially robust ← Most beneficial in optimizing RD ← Training on the selected data

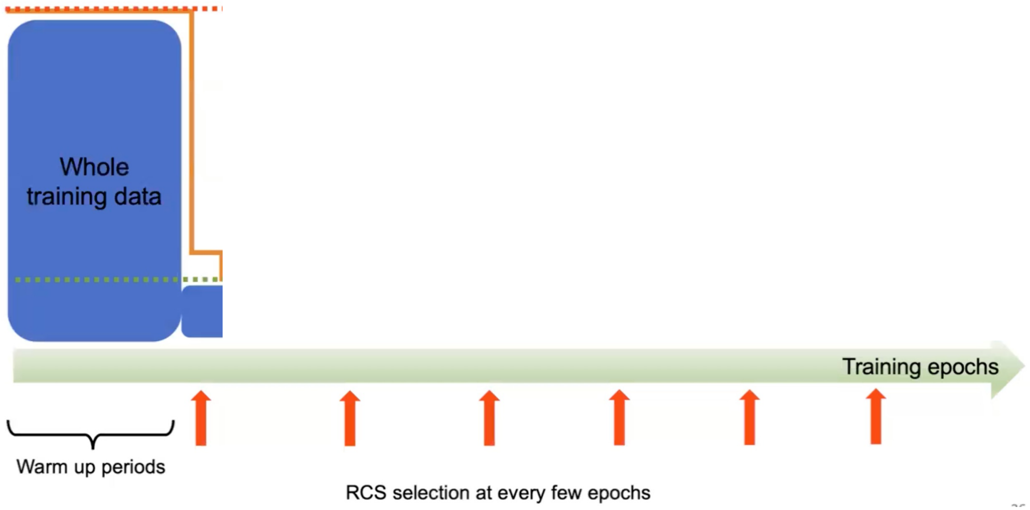
Algorithm: Efficient ACL via RCS

- Step 1 (Warm up): Warm up training on entire training set to find a better starting point f_θ.



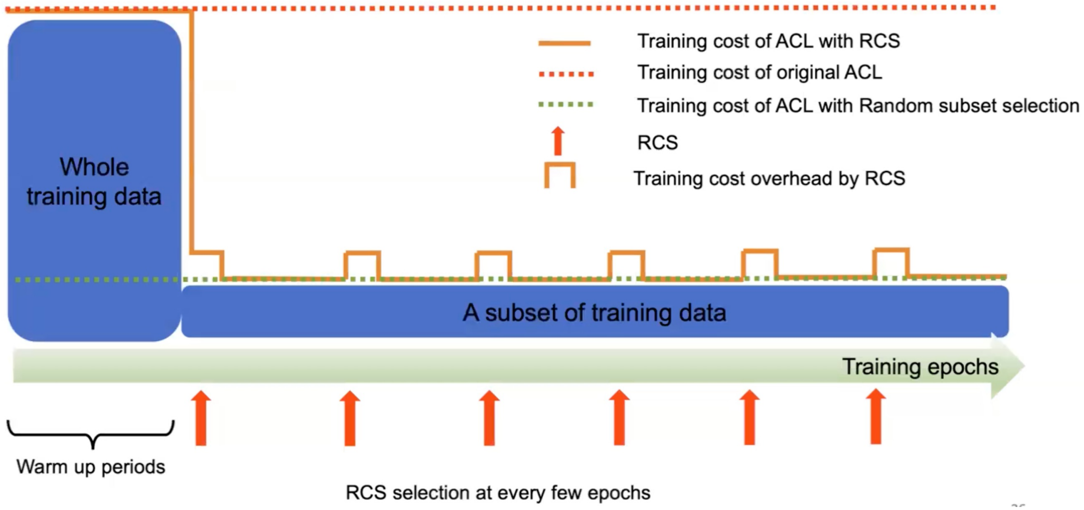
Algorithm: Efficient ACL via RCS

- Step 1 (Warm up): Warm up training on entire training set to find a better starting point f_θ.
- **Step 2.1 (RCS)**: $S \leftarrow \emptyset$. $\theta' \leftarrow \theta$. Compute gradients $Q \leftarrow \{q_k = \nabla_\theta \mathcal{L}_{\mathrm{ACL}}(x_k; \theta) \mid \forall x_k \in X\}$ on unlabeled training dataset X.
- **Step 2.2 (RCS)**: Compute gradients $q_U \leftarrow \nabla_\theta \mathcal{L}_{\mathrm{RD}}(U; \theta')$ on unlabeled validation dataset U.
- **Step 2.3 (RCS)**: Select a data x_k, whose gradient q_k matches best with q_U, i.e., $\arg\max_k \{q_k^\top q_U\}$. $G_\theta(x \mid S)$
- **Step 2.4 (RCS)**: $S \leftarrow S \cup \{x_k\}$, $X \leftarrow X \setminus \{x_k\}$, $\theta' \leftarrow \theta' - \eta' q_k$.
- **Step 2.5 (RCS)**: Repeat Step 2.2-2.3 until $|S| / |X| = k$.



Algorithm: Efficient ACL via RCS

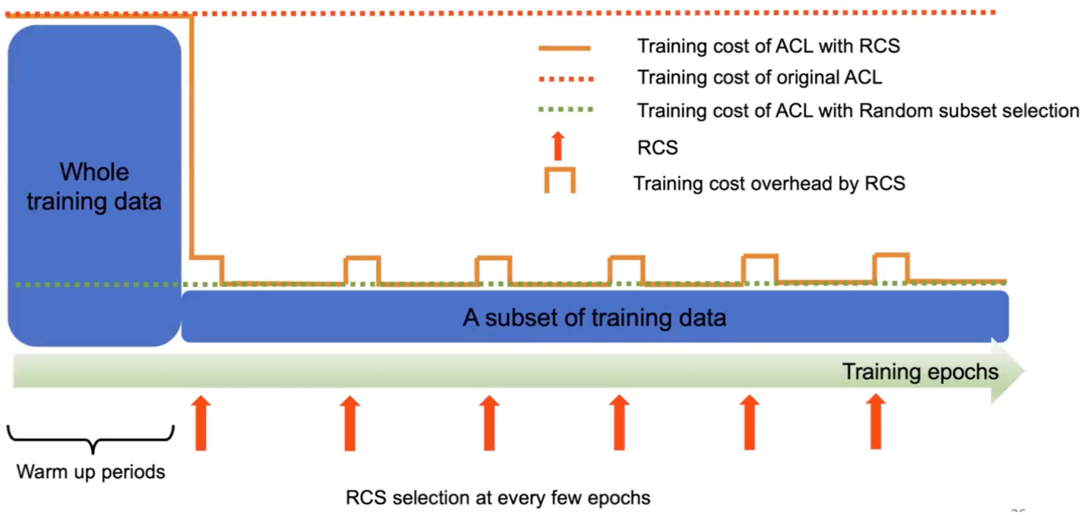
- Step 1 (Warm up): Warm up training on entire training set to find a better starting point f_θ.
- **Step 2.1 (RCS)**: $S \leftarrow \emptyset$. $\theta' \leftarrow \theta$. Compute gradients $Q \leftarrow \{q_k = \nabla_\theta \mathcal{L}_{\mathrm{ACL}}(x_k;\theta) \mid \forall x_k \in X\}$ on unlabeled training dataset X.
- **Step 2.2 (RCS)**: Compute gradients $q_U \leftarrow \nabla_\theta \mathcal{L}_{\mathrm{RD}}(U;\theta')$ on unlabeled validation dataset U.
- **Step 2.3 (RCS)**: Select a data x_k, whose gradient q_k matches best with q_U, i.e., $\arg\max_k \{q_k^\top q_U\}$. $G_\theta(x \mid S)$
- **Step 2.4 (RCS)**: $S \leftarrow S \cup \{x_k\}$, $X \leftarrow X \setminus \{x_k\}$, $\theta' \leftarrow \theta' - \eta' q_k$.
- **Step 2.5 (RCS)**: Repeat Step 2.2-2.3 until $\mid S \mid / \mid X \mid = k$.
- Step 3 (ACL training): Update parameters $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\mathrm{ACL}}(S;\theta)$
- Step 4: Every I epochs, go to Step 2.1 to generate a new coreset; otherwise go to Step 3 to update model parameters. The algorithm stops when reaches the final training epoch.



Algorithm: Efficient ACL via RCS

- Step 1 (Warm up): Warm up training on entire training set to find a better starting point f_θ.
- **Step 2.1 (RCS)**: $S \leftarrow \emptyset$. $\theta' \leftarrow \theta$. Compute gradients $Q \leftarrow \{q_k = \nabla_\theta \mathcal{L}_{\mathrm{ACL}}(x_k; \theta) \mid \forall x_k \in X\}$ on unlabeled training dataset X.
- **Step 2.2 (RCS)**: Compute gradients $q_U \leftarrow \nabla_\theta \mathcal{L}_{\mathrm{RD}}(U; \theta')$ on unlabeled validation dataset U.
- **Step 2.3 (RCS)**: Select a data x_k, whose gradient q_k matches best with q_U, i.e., $\arg\max_k \{q_k^\top q_U\}$. $G_\theta(x \mid S)$
- **Step 2.4 (RCS)**: $S \leftarrow S \cup \{x_k\}$, $X \leftarrow X \setminus \{x_k\}$, $\theta' \leftarrow \theta' - \eta' q_k$.
- **Step 2.5 (RCS)**: Repeat Step 2.2-2.3 until $\mid S \mid / \mid X \mid = k$.
- Step 3 (ACL training): Update parameters $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\mathrm{ACL}}(S; \theta)$
- Step 4: Every I epochs, go to Step 2.1 to generate a new coreset; otherwise go to Step 3 to update model parameters. The algorithm stops when reaches the final training epoch.

Whether the greedy search provide any optimality guarantee theoretically?



Theoretical analysis: greedy search has an optimality guarantee

Step 2.3 (RCS): Select a data $\boldsymbol{x}_k$, whose gradient $\boldsymbol{q}_k$ matches best with $\boldsymbol{q}_U$, i.e., $\arg\max_k \{q_k^\top q_U\}$. $G_\theta(x \mid S)$

- Proof sketch using a proxy set problem

$$\hat{S}^* = \underset{S \subseteq X, |S|/|X| = k}{\arg\max} \; \hat{G}_\theta(S) = \underset{S \subseteq X, |S|/|X| = k}{\arg\max} \; G_\theta(S) + |S|\sigma$$

1. Monotonicity

$\hat{G}(x \mid X) = \hat{G}(S \cup \{x\}) - \hat{G}(S) \geq 0$ for any $S \subseteq X$ and $x \in X \setminus S$.

More data, better representation in terms of robustness

2. γ^*-submodularity---diminishing returns

$$\underset{A,B \mid A \subseteq B}{\forall} G_\theta(x \mid A) \geq (1 - \gamma^*) G_\theta(x \mid B)$$

where $\gamma^* = \frac{1}{2\sigma - 1} \in (0, 1)$ and $A \subseteq B \subseteq X$

More data have diminishing gains for learning representations

Proxy coreset $\hat{S}^*$

Guaranteed lower bound of the original set problem using the proxy coreset

$$G_\theta(\hat{S}^*) \geq \boxed{G_\theta^* - (G_\theta^* + kN\sigma) \cdot e^{-\gamma^*}}$$

Proxy coreset selected via greedy search based on the proxy marginal gain $\hat{G}_\theta(x \mid S)$

Guaranteed Lower bound

Theoretical analysis: greedy search has an optimality guarantee

- Proof sketch using a proxy set problem $\hat{S}^* = \underset{S \subseteq X, |S|/|X|=k}{\arg\max} \ \hat{G}_\theta(S) = \underset{S \subseteq X, |S|/|X|=k}{\arg\max} \ G_\theta(S) + |S|\sigma$

Step 2.3 (RCS): Select a data $\boldsymbol{x}_k$, whose gradient $\boldsymbol{q}_k$ matches best with $\boldsymbol{q}_U$, i.e., $\arg\max_k \{q_k^\top q_U\}$. $G_\theta(x \mid S)$

$G_\theta(\hat{S}^*) \geq G_\theta^* - (G_\theta^* + kN\sigma) \cdot e^{-\gamma^*}$

Our greedy search via $\boxed{\text{the original marginal gain}}$ $\xrightarrow{\text{Approximate}}$ the proxy marginal gain $\xrightarrow{\hat{S}^*}$ The proxy coreset provides a guaranteed lower bound!

$$
\begin{aligned}
\hat{G}_\theta(x \mid S) &= \hat{G}_\theta(S \cup \{x\}) - \hat{G}_\theta(S) \\
&\approx \eta \nabla_\theta \mathcal{L}_{RD}(U; \theta - \eta \nabla_\theta \mathcal{L}_{\mathrm{ACL}}(S;\theta))^\top \nabla_\theta \mathcal{L}_{\mathrm{ACL}}(\{x\};\theta) + \sigma \\
&= \boxed{G_\theta(x \mid S)} + \sigma.
\end{aligned}
$$

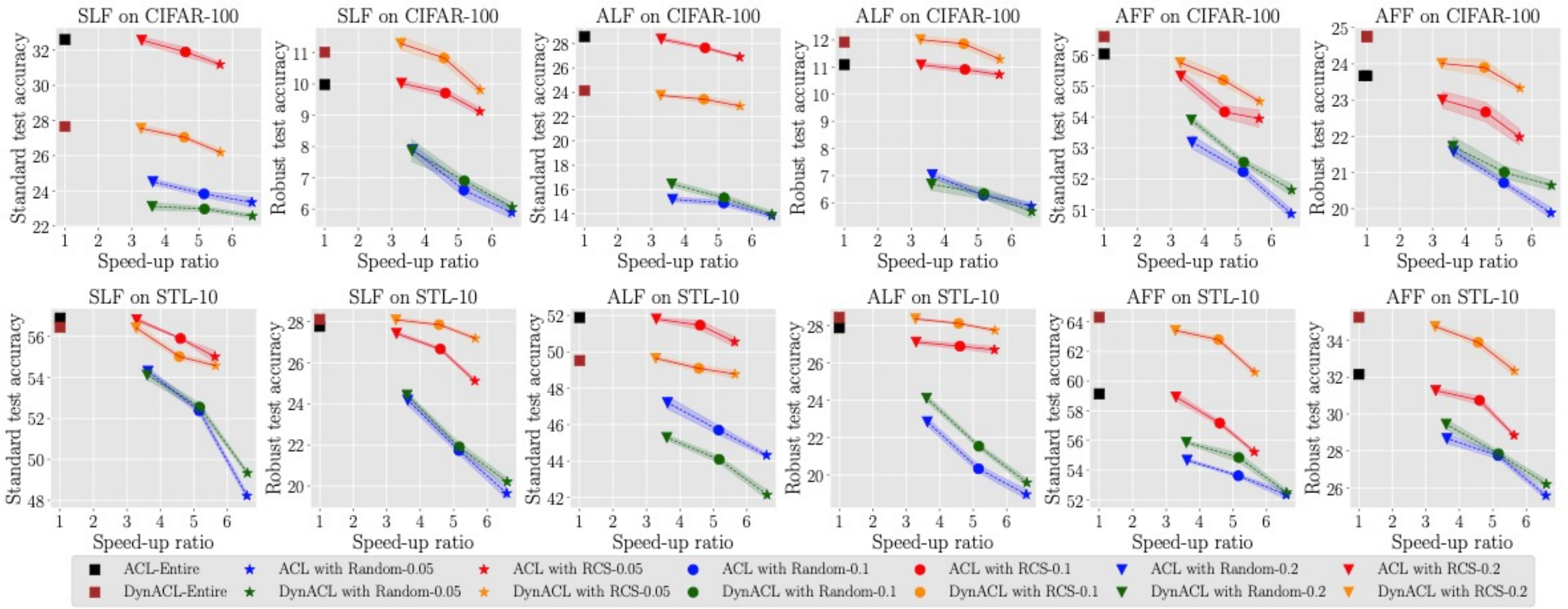
Can be discarded since it is a constant

Empirical Results

RCS is more efficient (higher speed-up ratio) compared to ACL on the entire set.

RCS is more effective (higher test accuracy) compared to random selection.

Effectiveness



Efficiency

The upper-right (ours) is better!

Empirical Results

For the first time to conduct ACL on ImageNet-1K using WRN-28-10

Table 1: Cross-task adversarial robustness transferability from ImageNet-1K to CIFAR-10.

Pre-training	Runing time (hours)	SLF		ALF		AFF	
		SA (%)	RA (%)	SA (%)	RA (%)	SA (%)	RA (%)
Standard CL	147.4	**84.36**±0.17	0.01±0.01	10.00±0.00	10.00±0.00	**86.63**±0.12	49.71±0.16
ACL on entire set	650.2	-	-	-	-	-	-
ACL with Random	94.3	68.75±0.06	15.89±0.06	59.57±0.28	27.14±0.19	84.75±0.18	50.12±0.21
ACL with RCS	111.8	70.02±0.12	**22.45**±0.13	**63.94**±0.21	**31.13**±0.17	85.23±0.23	**52.21**±0.14

Table 2: Cross-task adversarial robustness transferability from ImageNet-1K to CIFAR-100.

Pre-training	Runing time (hours)	SLF		ALF		AFF	
		SA (%)	RA (%)	SA (%)	RA (%)	SA (%)	RA (%)
Standard CL	147.4	**57.34**±0.23	0.01±0.01	9.32±0.01	0.06±0.01	**61.33**±0.12	25.11±0.15
ACL on entire set	650.2	-	-	-	-	-	-
ACL with Random	94.3	38.53±0.15	10.50±0.13	28.44±0.23	11.93±0.21	59.63±0.33	25.46±0.26
ACL with RCS	111.8	40.28±0.17	**14.55**±0.10	**33.15**±0.26	**14.89**±0.16	60.25±0.18	**28.24**±0.13

We prove the possibility of applying ACL on large-scale datasets.

Empirical Results

RCS for speeding up supervised adversarial training (SAT) on ImageNet-1K

while maintaining robustness transferability.

Table 17: Cross-task adversarial robustness transferability of adversarially pre-trained WRN-28-10 from ImageNet-1K to CIFAR-10. Here, "RA" stands for robust test accuracy under PGD-20 attacks following the setting of Hendrycks et al. [54]. The number after the dash line denotes subset fraction $k \in \{0.05, 0.1, 0.2\}$.

Pre-training	Runing time (hours)	ALF		AFF	
		SA (%)	RA (%)	SA (%)	RA (%)
Standard training on entire set	66.7	10.12	10.04	84.73	51.91
SAT [54] on entire set	341.7	85.90	50.89	89.35	59.68
SAT with Random-0.05	53.6	69.59	31.58	85.55	53.53
SAT with RCS-0.05	**68.6**	**79.72**	**44.44**	**87.99**	**56.87**
SAT with Random-0.1	70.2	73.28	33.86	86.78	54.95
SAT with RCS-0.1	**81.9**	**81.92**	**45.10**	**88.87**	**57.69**
SAT with Random-0.2	103.4	75.46	39.62	86.64	56.46
SAT with RCS-0.2	**121.9**	**83.94**	**46.88**	**89.54**	**58.13**

RCS is also applicable to robust supervised pre-training!

Towards Robust Foundation Models

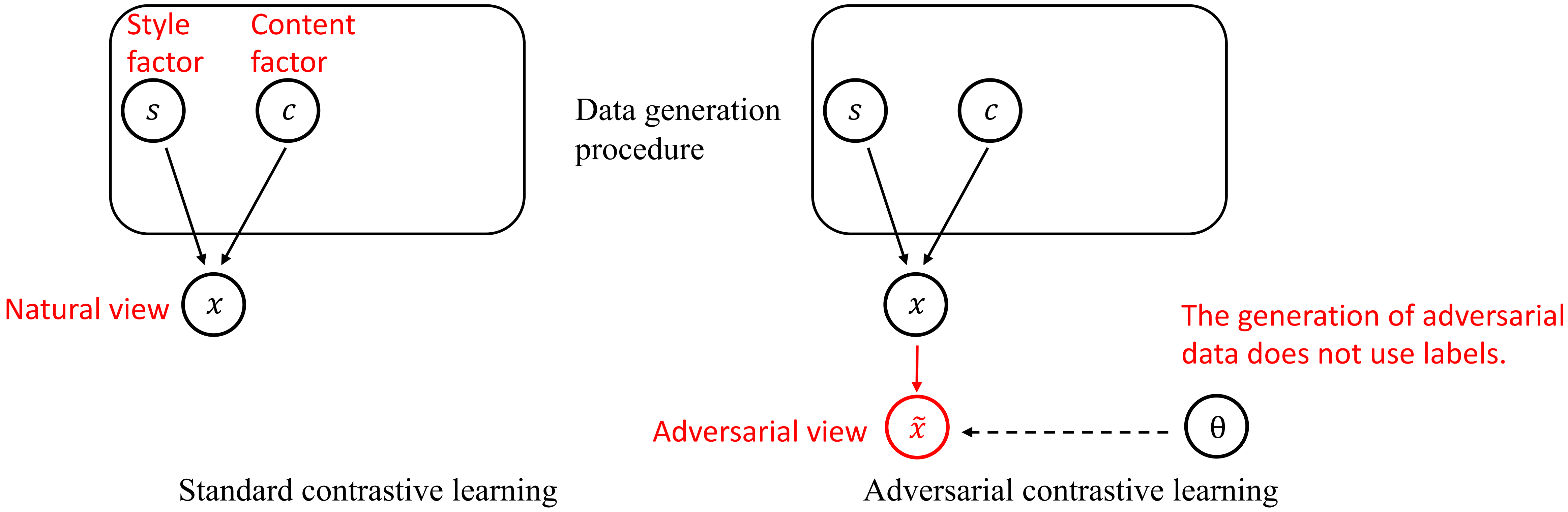
Efficient and Effective Adversarial Contrastive Learning

Motivation

- *Limited robustness transferability* to downstream tasks

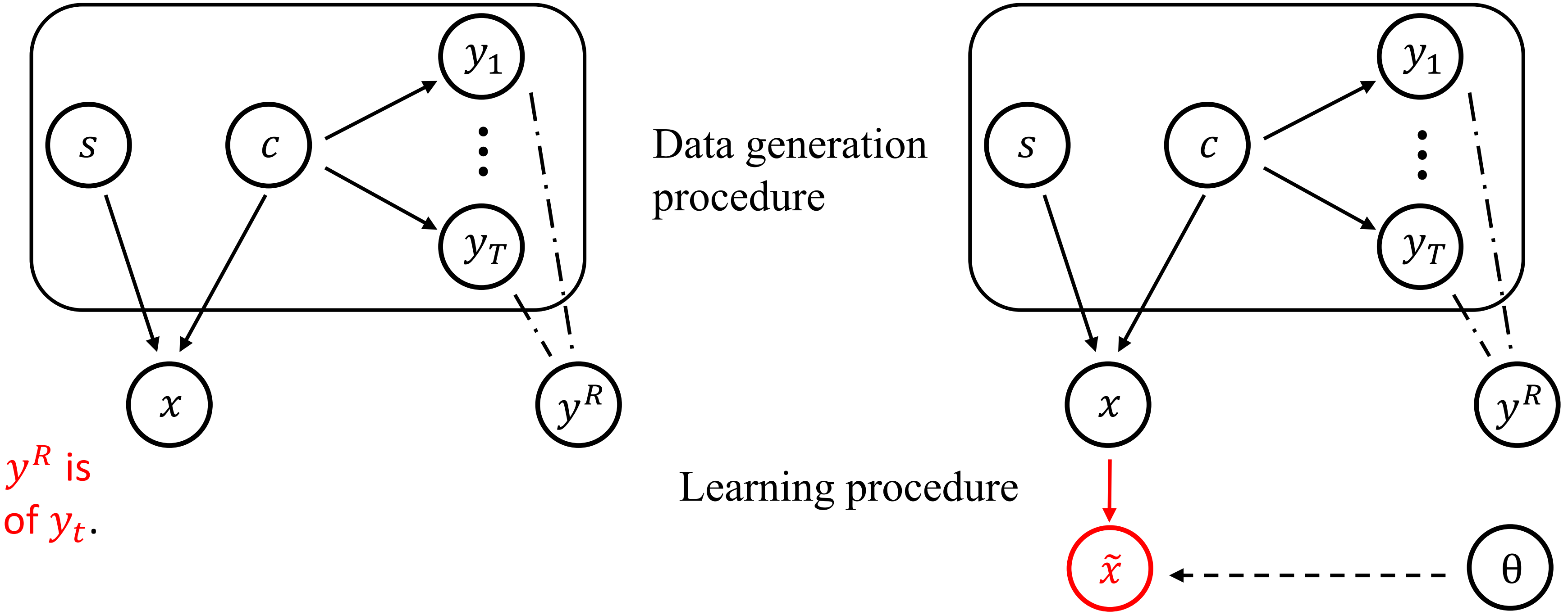
How can we improve ACL's robustness transferability?

Causal View of ACL



Causal View of ACL

Target labels from an
unknown downstream task



The proxy label y^R is
the refinement of y_t.

Standard contrastive learning

Adversarial contrastive learning

Proxy label y^R

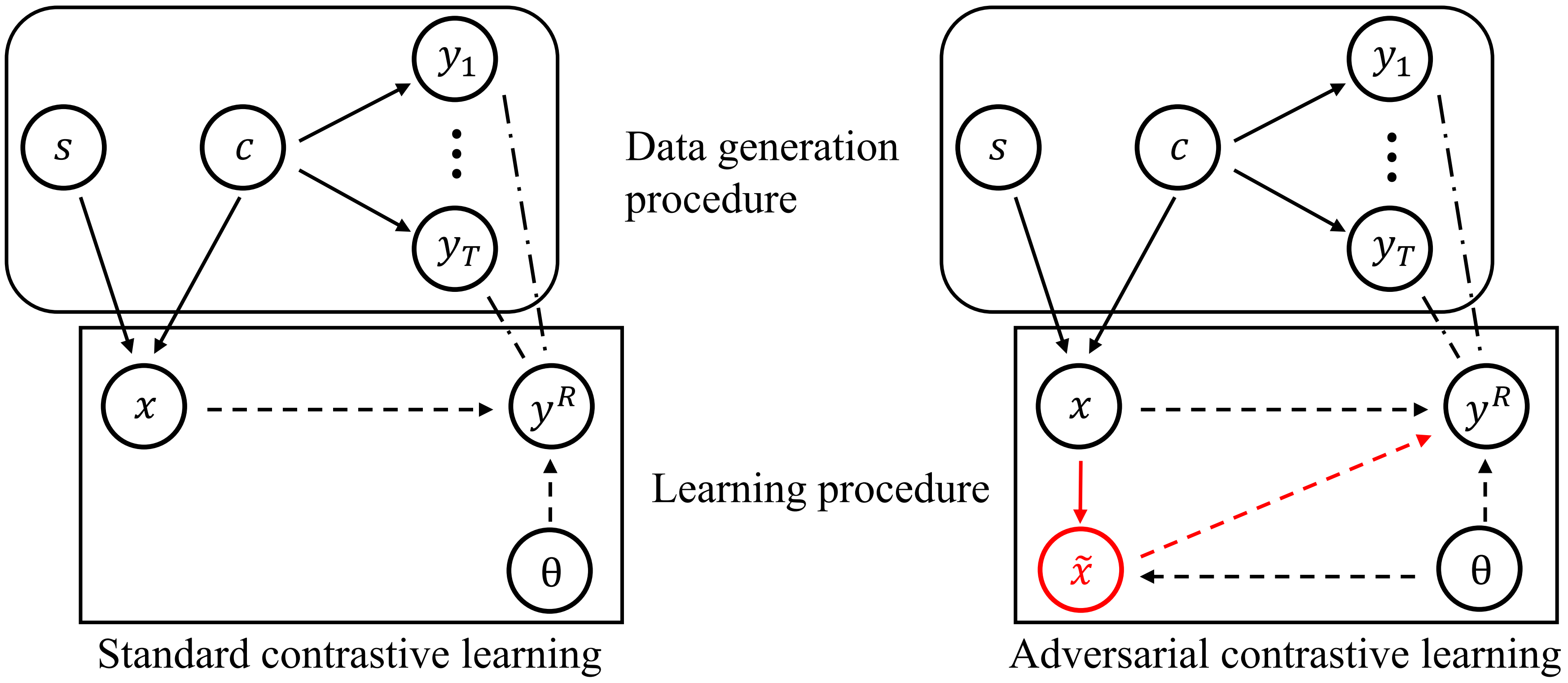
Target label y_t

Golden Retriever
with *yellow hair*

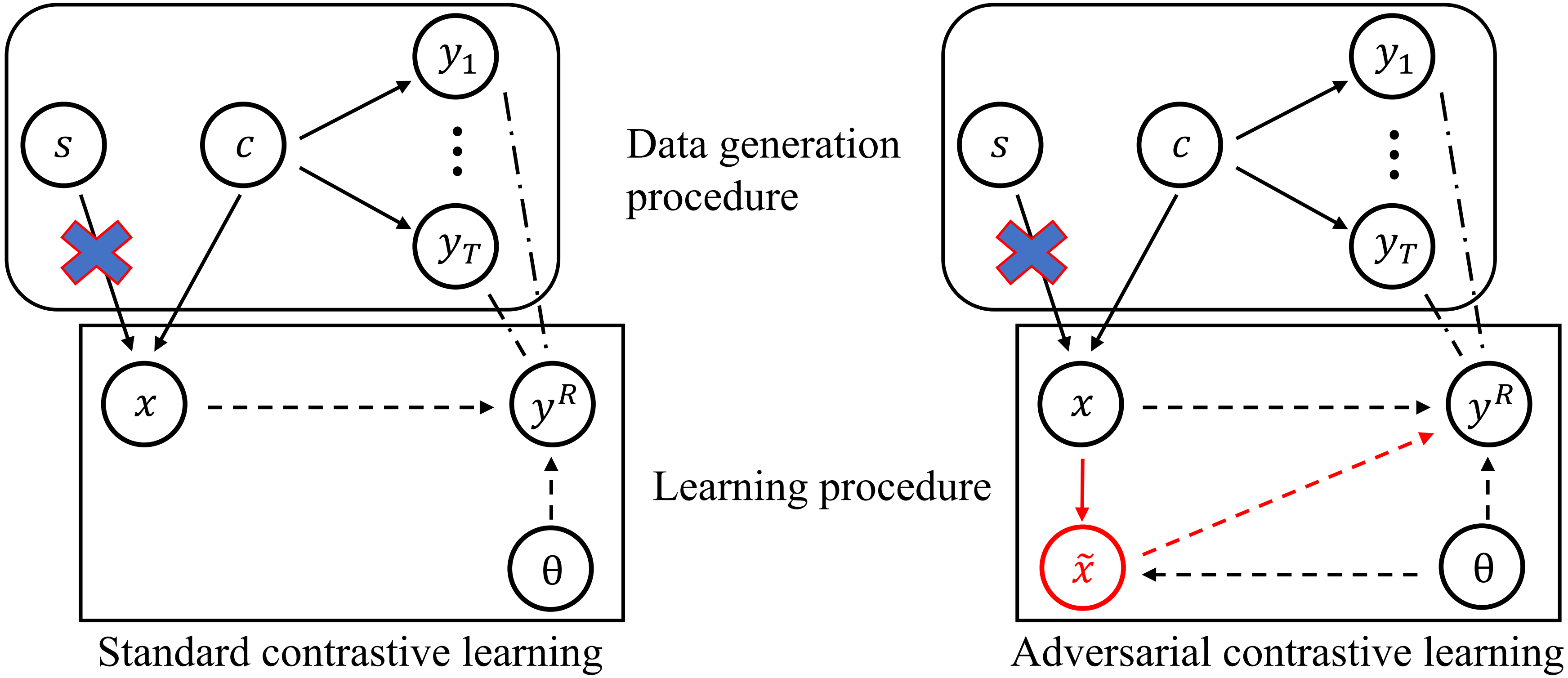
Dog

Labrador Retriever
with *black hair*

Causal View of ACL



Causal View of ACL



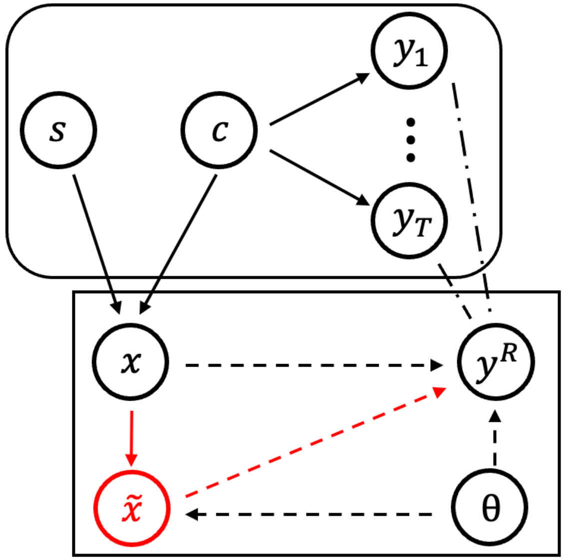
Style-invariant criterion: The intervention on the style factor should not affect the conditional probability

$$p^{\boxed{do(\tau_i)}}(y^R \mid x) = p^{do(\tau_j)}(y^R \mid x)$$

Adversarial Invariant Regularization (AIR)

- The conditional probability learned via ACL the mild assumption of Markov condition

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

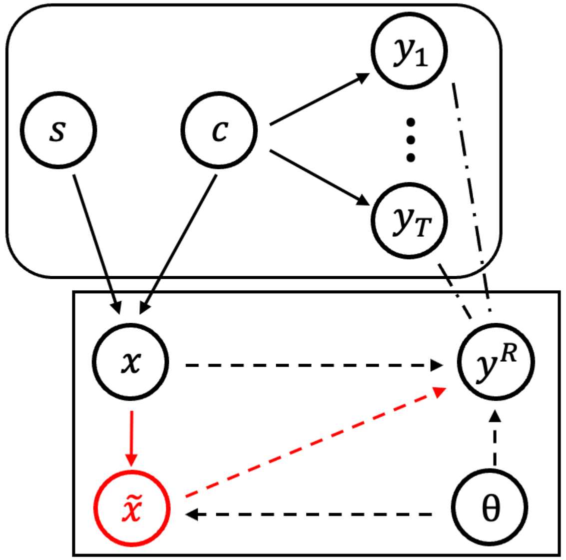


Adversarial contrastive learning

Adversarial Invariant Regularization (AIR)

- The conditional probability learned via ACL $p(y^R|x) = p(y^R|\tilde{x})p(\tilde{x}|x)$
- Style-independent criterion: The intervention on the style factor should not affect the conditional probability

$$p^{do(\tau_i)}(y^R \mid x) = p^{do(\tau_j)}(y^R \mid x)$$

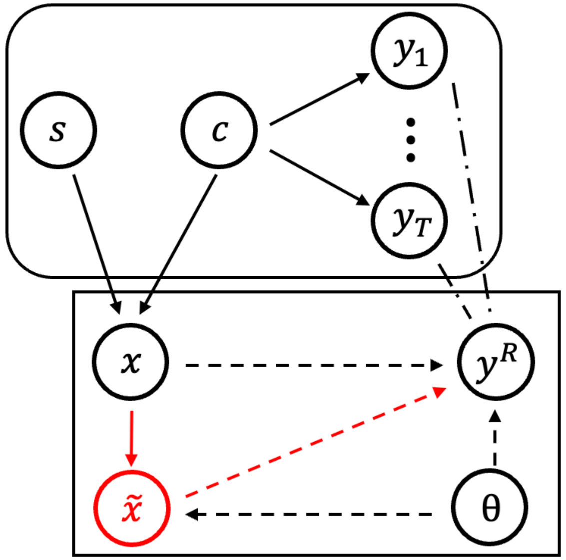


Adversarial contrastive learning

Adversarial Invariant Regularization (AIR)

- The conditional probability learned via ACL $p(y^R|x) = p(y^R|\tilde{x})p(\tilde{x}|x)$
- Style-independent criterion: The intervention on the style factor should not affect the conditional probability

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



Adversarial contrastive learning

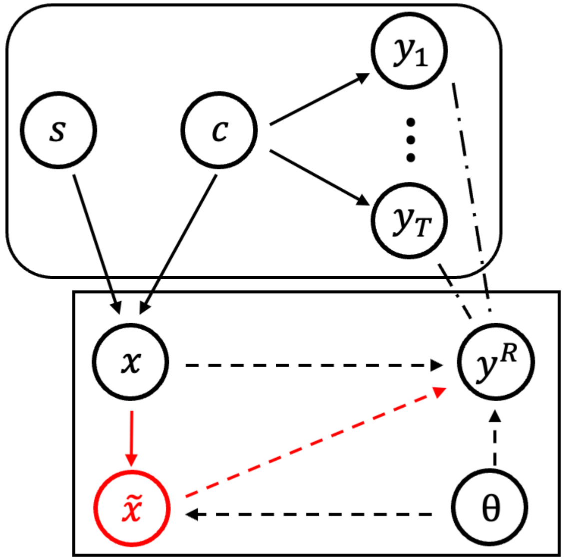
Adversarial Invariant Regularization (AIR)

- The conditional probability learned via ACL $p(y^R|x) = p(y^R|\tilde{x})p(\tilde{x}|x)$
- Style-independent criterion: The intervention on the style factor should not affect the conditional probability

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

- Loss function of AIR -> to enforce style-independence

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



Adversarial contrastive learning

Understanding of AIR

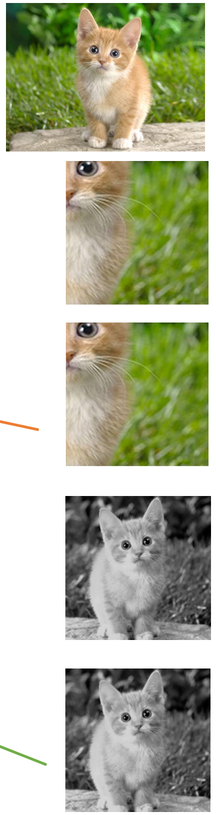
$$p^{do(\tau_i)}(y^R|\tilde{x})p^{do(\tau_i)}(\tilde{x}|x)$$

$$\mathcal{L}_{\text{AIR}}(B;\theta,\epsilon)$$

$$p^{do(\tau_j)}(y^R|\tilde{x})p^{do(\tau_j)}(\tilde{x}|x)$$

$$p^{do(\tau_u)}(y^R|\tilde{x}) = \frac{e^{\text{sim}(f_\theta(x), f_\theta(\tilde{x}^u))/t}}{\sum_{x_k \in B} e^{\text{sim}\left(f_\theta(x_k), f_\theta(\tilde{x}_k^u)\right)/t}}$$

the self-consistency between
the original view and
the adversarial view under



x_k original view

x_k^i

$\tilde{x}_k^i$ adversarial view_i

x_k^j

$\tilde{x}_k^j$ adversarial view_j

Understanding of AIR

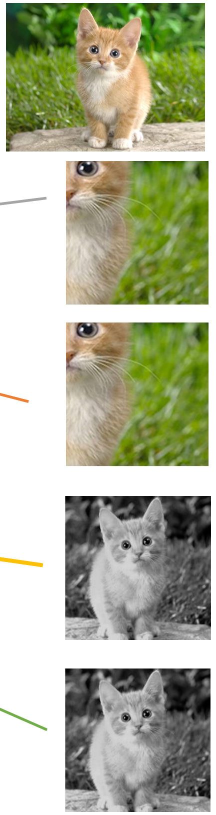
$$p^{do(\tau_i)}(y^R|\tilde{x})\boxed{p^{do(\tau_i)}(\tilde{x}|x)}$$

$$\mathcal{L}_{\text{AIR}}(B;\theta,\epsilon)$$

$$p^{do(\tau_j)}(y^R|\tilde{x})\boxed{p^{do(\tau_j)}(\tilde{x}|x)}$$

$$p^{do(\tau_u)}(\tilde{x}|x) = \frac{e^{\text{sim}(f_\theta(\tilde{x}^u), f_\theta(x^u))/t}}{\sum_{x_k \in B} e^{\text{sim}\left(f_\theta(\tilde{x}_k^u), f_\theta(x_k^u)\right)/t}}$$

the self-consistency between
the adversarial view and
the natural view



x_k

x_k^i natural view_i

$\tilde{x}_k^i$ adversarial view_i

x_k^j natural view_j

$\tilde{x}_k^j$ adversarial view_j

Understanding of AIR

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

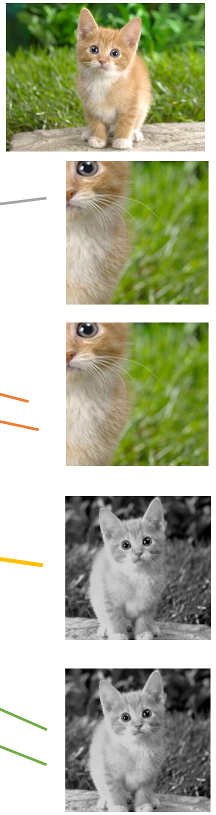
AIR enforces self-consistency
to be cross-consistent
under different augmentations

$$\mathcal{L}_{\mathrm{AIR}}(B;\theta,\epsilon)$$

$$p^{do(\tau_i)}(y^R|\tilde{x})p^{do(\tau_i)}(\tilde{x}|x)$$

$$p^{do(\tau_j)}(y^R|\tilde{x})p^{do(\tau_j)}(\tilde{x}|x)$$

the self-consistency
among the different views



x_k original view

x_k^i natural view_i

$\tilde{x}_k^i$ adversarial view_i

x_k^j natural view_j

$\tilde{x}_k^j$ adversarial view_j

Algorithm: Enhancing ACL via AIR

Algorithm 1 ACL with Adversarial Invariant Regularization (AIR)

Input: Unlabeled training set U, total training epochs E, learning rate η, batch size β, adversarial budget $\epsilon > 0$, hyperparameters λ_1 and λ_2
Output: Pre-trained representation extractor h_θ
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 $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \sum_{x_k \in B_m} \ell_{\mathrm{ACL}}(x_k; \theta) + \boxed{\lambda_1 \cdot \mathcal{L}_{\mathrm{AIR}}(B_m; \theta, 0) + \lambda_2 \cdot \mathcal{L}_{\mathrm{AIR}}(B_m; \theta, \epsilon)}$
  end for
end for

$\epsilon > 0$:Regulate robust representations

$\epsilon = 0$: Regulate standard representations

AIR achieves SOTA robustness transferability

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

$\mathcal{D}_1 \to \mathcal{D}_2$	Pre-training	SLF AA (%)	SLF SA (%)	ALF AA (%)	ALF SA (%)	AFF AA (%)	AFF SA (%)
CIFAR-10 → CIFAR-100	ACL [26]	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 [33]	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 [26]	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 [33]	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 [26]	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 [33]	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 [26]	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 [33]	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

AIR achieves SOTA robustness transferability

- Robustness transferability via automated robust fine-tuning
  - AutoLoRa: an automated and parameter-free robust fine-tuning framework

Table 7: Cross-task adversarial robustness transferability evaluated via AutoLoRa [44]. $\mathcal{D}_1 \to \mathcal{D}_2$ 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).

$\mathcal{D}_1 \to \mathcal{D}_2$	Finetuning mode	Pre-training	AutoLoRa [44] AA (%)	AutoLoRa [44] SA (%)	Diff AA (%)	Diff SA (%)
CIFAR-10 $\to$ STL-10	SLF	DynACL [33]	30.18	54.23	+1.01	+1.82
		DynACL-AIR	**30.48**	**56.56**	**+0.84**	**+0.72**
	ALF	DynACL [33]	31.72	57.30	+2.13	+7.75
		DynACL-AIR	**31.81**	**57.40**	**+0.57**	**+0.26**
	AFF	DynACL [33]	35.51	64.16	+0.26	+0.63
		DynACL-AIR	**35.88**	**64.25**	**+0.22**	**+0.51**
CIFAR-100 $\to$ STL-10	SLF	DynACL [33]	23.27	48.93	+0.10	+1.39
		DynACL-AIR	**23.44**	**50.28**	**+0.20**	**+2.08**
	ALF	DynACL [33]	26.53	48.56	+0.29	+2.86
		DynACL-AIR	**26.89**	**49.02**	**+0.29**	**+0.47**
	AFF	DynACL [33]	31.25	58.56	+0.08	+0.06
		DynACL-AIR	**31.57**	**58.65**	**+0.15**	**+0.21**

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

AIR ranks First in RobustSSL Benchmark

Standard Linear Fine-Tuning (SLF)			Vanilla Fine-Tuning		
Rank	**Paper**	**Venue**	**Robust Accuracy**	**Corruption Accuracy**	**Standard Accuracy**
1	Enhancing Adversarial Contrastive Learning via Adversarial Invariant Regularization *Using post-processing	*NeurIPS 2023*	**46.99**	72.11	81.80
2	Rethinking the Effect of Data Augmentation in Adversarial Contrastive Learning *Using post-processing	*ICLR 2023*	**46.54**	71.96	79.82
3	Enhancing Adversarial Contrastive Learning via Adversarial Invariant Regularization	*NeurIPS 2023*	**45.17**	70.51	78.08
4	Rethinking the Effect of Data Augmentation in Adversarial Contrastive Learning	*ICLR 2023*	**45.09**	68.67	75.41
5	Efficient Adversarial Contrastive Learning via Robustness-Aware Coreset Selection	*NeurIPS 2023*	**44.29**	69.56	77.14
6	Decoupled Adversarial Contrastive Learning for Self-supervised Adversarial Robustness	*ECCV 2022*	**43.27**	73.06	79.94
7	When Does Contrastive Learning Preserve Adversarial Robustness from Pretraining to Finetuning? *Using ImageNet-1K pre-trained models	*NeurIPS 2021*	**43.18**	73.14	82.36
8	Adversarial Contrastive Learning via Asymmetric InfoNCE *Using ImageNet-1K pre-trained models	*ECCV 2022*	**42.72**	74.09	83.70
9	Robust Pre-Training by Adversarial Contrastive Learning	*NeurIPS 2020*	**39.17**	70.72	78.22
10	Adversarial Self-Supervised Contrastive Learning	*NeurIPS 2020*	**26.12**	-	77.90

Adversarial Linear Fine-Tuning (ALF)			Vanilla Fine-Tuning		
Rank	**Paper**	**Venue**	**Robust Accuracy**	**Corruption Accuracy**	**Standard Accuracy**
1	Enhancing Adversarial Contrastive Learning via Adversarial Invariant Regularization *Using post-processing	*NeurIPS 2023*	**48.23**	71.74	79.56
2	Rethinking the Effect of Data Augmentation in Adversarial Contrastive Learning *Using post-processing	*ICLR 2023*	**47.98**	70.89	78.81
3	Enhancing Adversarial Contrastive Learning via Adversarial Invariant Regularization	*NeurIPS 2023*	**46.14**	69.97	77.42
4	Efficient Adversarial Contrastive Learning via Robustness-Aware Coreset Selection	*NeurIPS 2023*	**45.75**	67.84	74.95
5	Rethinking the Effect of Data Augmentation in Adversarial Contrastive Learning	*ICLR 2023*	**45.67**	66.69	72.97
6	When Does Contrastive Learning Preserve Adversarial Robustness from Pretraining to Finetuning? *Using ImageNet-1K pre-trained models	*NeurIPS 2021*	**44.05**	71.50	80.04
7	Adversarial Contrastive Learning via Asymmetric InfoNCE *Using ImageNet-1K pre-trained models	*ECCV 2022*	**43.28**	71.61	80.30
8	Decoupled Adversarial Contrastive Learning for Self-supervised Adversarial Robustness	*ECCV 2022*	**41.99**	71.66	77.71
9	Robust Pre-Training by Adversarial Contrastive Learning	*NeurIPS 2020*	**40.60**	68.56	75.53
10	Adversarial Self-Supervised Contrastive Learning	*NeurIPS 2020*	**29.69**	-	75.62

Robust Self-Supervised Learning (RobustSSL) Benchmark

https://robustssl.github.io

Thank you for your attention!

- Summary
  - With RCS and AIR, we can efficiently build effective robust foundation models!
- Potential future directions
  - Explore the potential applications of ACL in various CV, NLP, and multi-modal tasks.
- References

1. Xu, Xilie, Jingfeng Zhang, Feng Liu, Masashi Sugiyama, and Mohan Kankanhalli. "Efficient Adversarial Contrastive Learning via Robustness-Aware Coreset Selection." NeurIPS 2023 (spotlight).
2. Xu, Xilie, Jingfeng Zhang, Feng Liu, Masashi Sugiyama, and Mohan Kankanhalli. "Enhancing Adversarial Contrastive Learning via Adversarial Invariant Regularization." NeurIPS 2023.
3. Xu, Xilie, Jingfeng Zhang, and Mohan Kankanhalli. "Autolora: A parameter-free automated robust fine-tuning framework." arXiv preprint arXiv:2310.01818 (2023).

- Q&A