Can contrastive learning avoid shortcut solutions?

Joshua Robinson[†] joshrob@mit.edu
Li Sun[*] lis118@pitt.edu
Ke Yu[*] yu.ke@pitt.edu
Kayhan Batmanghelich[*] kayhan@pitt.edu
Stefanie Jegelka[†] stefje@csail.mit.edu
Suvrit Sra[†] suvrit@mit.edu
[†]*Massachusetts Institute of Technology*
[*]*University of Pittsburgh*

arXiv:2106.11230v1 [cs.LG] 21 Jun 2021

https://github.com/joshr17/IFM

Problem

- **Generalization of representations learned via contrastive learning depends on what features of the data are extracted.**
- **Contrast loss does not always sufficiently guide which features are extracted, which may suppress important predictive features.**

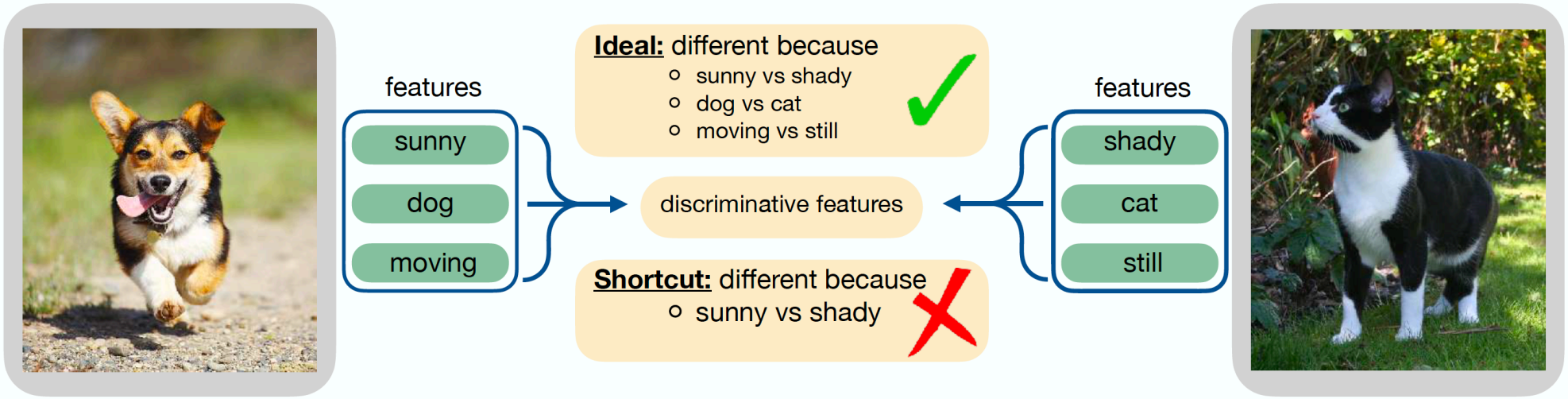

Figure 1: An ideal encoder would discriminate between instances using multiple distinguishing features instead of finding simple shortcuts that suppress features. We show that InfoNCE-trained encoders can suppress features (Sec. 2.2). However, making instance discrimination harder during training can trade off representation of different features (Sec. 2.3). To avoid the need for trade-offs we propose *implicit feature modification* (Sec. 3), which reduces suppression in general, and improves generalization (Sec. 4).

Definition of feature suppression

Formally, we assume that the data has underlying feature spaces $\mathcal{Z}^1, \ldots, \mathcal{Z}^n$ with a distribution p_j on each $\mathcal{Z}^j$. Each $j \in [n]$, corresponding to a latent space $\mathcal{Z}^j$, models a distinct feature. We write the product as $\mathcal{Z}^S = \prod_{j \in S} \mathcal{Z}^j$, and simply write $\mathcal{Z}$ instead of $\mathcal{Z}^{[n]}$ where $[n] = \{1, \ldots, n\}$. A set of features $z = (z^j)_{j \in [n]} \in \mathcal{Z}$ is generated by sampling each coordinate $z^j \in \mathcal{Z}^j$ independently, and we denote the measure on $\mathcal{Z}$ induced by z by λ. Further, let $\lambda(\cdot|z^S)$ denote the conditional measure on $\mathcal{Z}$ for fixed z^S. For $S \subseteq [n]$ we use z^S to denote the projection of z onto $\mathcal{Z}^S$. Finally, an injective map $g : \mathcal{Z} \to \mathcal{X}$ produces observations $x = g(z)$.

Our aim is to train an encoder $f : \mathcal{X} \to \mathbb{S}^{d-1}$ to map input data x to the surface of the unit sphere $\mathbb{S}^{d-1} = \{u \in \mathbb{R}^d : \|u\|_2 = 1\}$ in such a way that f extracts useful information. To formally define feature suppression, we need the *pushforward* $h\#\nu(V) = \nu(h^{-1}(V))$ of a measure ν on a space $\mathcal{U}$ for a measurable map $h : \mathcal{U} \to \mathcal{V}$ and measurable $V \subseteq \mathcal{V}$, where $h^{-1}(V)$ denotes the preimage.

Definition 1. Consider an encoder $f : \mathcal{X} \to \mathbb{S}^{d-1}$ and features $S \subseteq [n]$. For each $z^S \in \mathcal{Z}^S$, let $\mu(\cdot|z^S) = (f \circ g)\#\lambda(\cdot|z^S)$ be the pushforward measure on $\mathbb{S}^{d-1}$ by $f \circ g$ of the conditional $\lambda(\cdot|z^S)$.

1. *f suppresses S* if for any pair $z^S, \bar{z}^S \in \mathcal{Z}^S$, we have $\mu(\cdot|z^S) = \mu(\cdot|\bar{z}^S)$.
2. *f distinguishes S* if for any pair of distinct $z^S, \bar{z}^S \in \mathcal{Z}^S$, measures $\mu(\cdot|z^S), \mu(\cdot|\bar{z}^S)$ have disjoint support.

Why optimizing the InfoNCE loss can still lead to feature suppression

$$\mathcal{L}_m(f) = \mathbb{E}_{x,x^+,\{x_i^-\}_{i=1}^m}\left[-\log \frac{e^{f(x)^\top f(x^+)/\tau}}{e^{f(x)^\top f(x^+)/\tau} + \sum_{i=1}^m e^{f(x)^\top f(x_i^-)/\tau}}\right], \tag{1}$$

loss that do and solutions that do not suppress a given feature. Following previous work [40, 49, 56], we analyze the loss as the number of negatives goes to infinity,

$$\mathcal{L} = \lim_{m\to\infty}\left\{\mathcal{L}_m(f) - \log m - \tfrac{2}{\tau}\right\} = \tfrac{1}{2\tau}\mathbb{E}_{x,x^+}\|f(x) - f(x^+)\|^2 + \mathbb{E}_{x^+}\log\left[\mathbb{E}_{x^-}e^{f(x^+)^\top f(x^-)/\tau}\right].$$

We subtract $\log m$ to ensure the limit is finite, and use x^- to denote a random sample with the same distribution as x_i^-. Prop. 1 (proved in App. A) shows that, assuming the marginals p_j are uniform, the InfoNCE loss is optimized both by encoders that suppress feature j, and by encoders that distinguish j.

Proposition 1. *Suppose that p_j is uniform on $\mathcal{Z}^j = \mathbb{S}^{d-1}$ for all $j \in [n]$. Then for any feature $j \in [n]$ there exists an encoder f_{supp} that suppresses feature j and encoder f_{disc} that discriminates j but both attain $\min_{f:\,measurable}\mathcal{L}(f)$.*

Controlling feature learning via the difficulty of instance discrimination

$$\mathcal{L}_m(f) = \mathbb{E}_{x,x^+,\{x_i^-\}_{i=1}^m}\left[-\log \frac{e^{f(x)^\top f(x^+)/\tau}}{e^{f(x)^\top f(x^+)/\tau} + \sum_{i=1}^m e^{f(x)^\top f(x_i^-)/\tau}}\right], \tag{1}$$

1. Temperature τ in the InfoNCE loss (Eqn. 1). Smaller τ places higher importance on positive an negative pairs with high similarity [47].
2. Hard negative sampling method of Robinson et al. [40], which uses importance sampling to sample harder negatives. The method introduces a hardness concentration parameter β, with larger β corresponding to harder negatives (see [40] for full details).

40. Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *Int. Conf. on Learning Representations (ICLR)*, 2021.

Proposition 2 (Informal). *Suppose that p_j is uniform on $\mathcal{Z}^j = \mathbb{S}^{d-1}$ for all $j \in [n]$. Further, for $S \subseteq [n]$ suppose that $x, x^+, \{x_i^-\}_i$ are conditioned on the event that they have the same features S. Then any f that minimizes the (limiting) InfoNCE loss suppresses features S.*

Implicit feature modification for reducing feature suppression

To avoid this effect, we develop a technique that *adaptively* modifies samples to remove whichever features are used to discriminate a particular positive pair from negatives, then trains an encoder to discriminate instances using *both* the original features, and the features left over after modification. While a natural method for modifying features is to directly transform raw input data, it is very challenging to modify the semantics of an input in this way. So instead we propose modifying features by applying transformations to encoded samples $v = f(x)$. Since we modify the encoded samples, instead of raw inputs x, we describe our method as *implicit*.

We set up our notation. Given batch $x, x^+, \{x_i^-\}_{i=1}^m$ we write $v = f(x)$, $v^+ = f(x^+)$, and $v_i^- = f(x_i^-)$ to denote the corresponding embeddings. As in Eqn. 1, the point-wise InfoNCE loss is,

$$\ell(v, v^+, \{v_i^-\}_{i=1}^m) = -\log \frac{e^{v^\top v^+/\tau}}{e^{v^\top v^+/\tau} + \sum_{i=1}^m e^{v^\top v_i^-/\tau}}.$$

Definition 2 (Implicit feature modification). Given budget $\boldsymbol{\varepsilon} \in \mathbb{R}_+^m$, and encoder $f : \mathcal{X} \to \mathbb{S}^d$, an adversary removes features from f that discriminates batch $x, x^+, \{x_i^-\}_{i=1}^m$ by maximizing the point-wise InfoNCE loss, $\ell_{\boldsymbol{\varepsilon}}(v, v^+, \{v_i^-\}_{i=1}^m) = \max_{\delta^+ \in \mathcal{B}_{\varepsilon^+}, \{\delta_i^- \in \mathcal{B}_{\varepsilon_i}\}_{i=1}^m} \ell(v, v^+ + \delta^+, \{v_i^- + \delta_i^-\}_{i=1}^m)$.

Implicit feature modification for reducing feature suppression

Lemma 1. *For any* $v, v^+, \{v_i^-\}_{i=1}^m \in \mathbb{R}^d$ *we have,*

$$\nabla_{v_j^-}\ell = \frac{e^{v^\top v_j^-/\tau}}{e^{v^\top v^+/\tau} + \sum_{i=1}^m e^{v^\top v_i^-/\tau}} \cdot \frac{v}{\tau} \quad \text{and} \quad \nabla_{v^+}\ell = \left(\frac{e^{v^\top v^+/\tau}}{e^{v^\top v^+/\tau} + \sum_{i=1}^m e^{v^\top v_i^-/\tau}} - 1\right) \cdot \frac{v}{\tau}.$$

In particular, $\nabla_{v_j^-}\ell \propto v$ *and* $\nabla_{v^+}\ell \propto -v$.

This expression shows that the adversary perturbs v_j^- (resp. v^+) in the direction of the anchor v (resp $-v$). Since the derivative directions are *independent* of $\{v_i^-\}_{i=1}^m$ and v^+, we can analytically compute optimal perturbations in $\mathcal{B}_\varepsilon$. Indeed, following the constant ascent direction shows the optimal updates are simply $v_i^- \leftarrow v_i^- + \varepsilon_i v$ and $v^+ \leftarrow v^+ - \varepsilon^+ v$. The positive (resp. negative) perturbations increase (resp. decrease) cosine similarity to the anchor $\text{sim}(v, v_i^- + \varepsilon_i v) \to 1$ as $\varepsilon_i \to \infty$ (resp. $\text{sim}(v, v^+ - \varepsilon^+ v) \to -1$ as $\varepsilon^+ \to \infty$). In Fig. 4 we visualize the newly synthesized v_i^-, v^+ and find meaningful interpolation of semantics. Plugging the update rules for v^+ and v_i^- into the point-wise InfoNCE loss yields,

$$\ell_{\boldsymbol{\varepsilon}}(v, v^+, \{v_i^-\}_{i=1}^m) = -\log \frac{e^{(v^\top v^+ - \varepsilon^+)/\tau}}{e^{(v^\top v^+ - \varepsilon^+)/\tau} + \sum_{i=1}^m e^{(v^\top v_i^- + \varepsilon_i)/\tau}}. \tag{2}$$

$$\min_f \{\mathcal{L}(f) + \alpha \mathcal{L}_{\boldsymbol{\varepsilon}}(f)\}/2$$

Implicit feature modification for reducing feature suppression

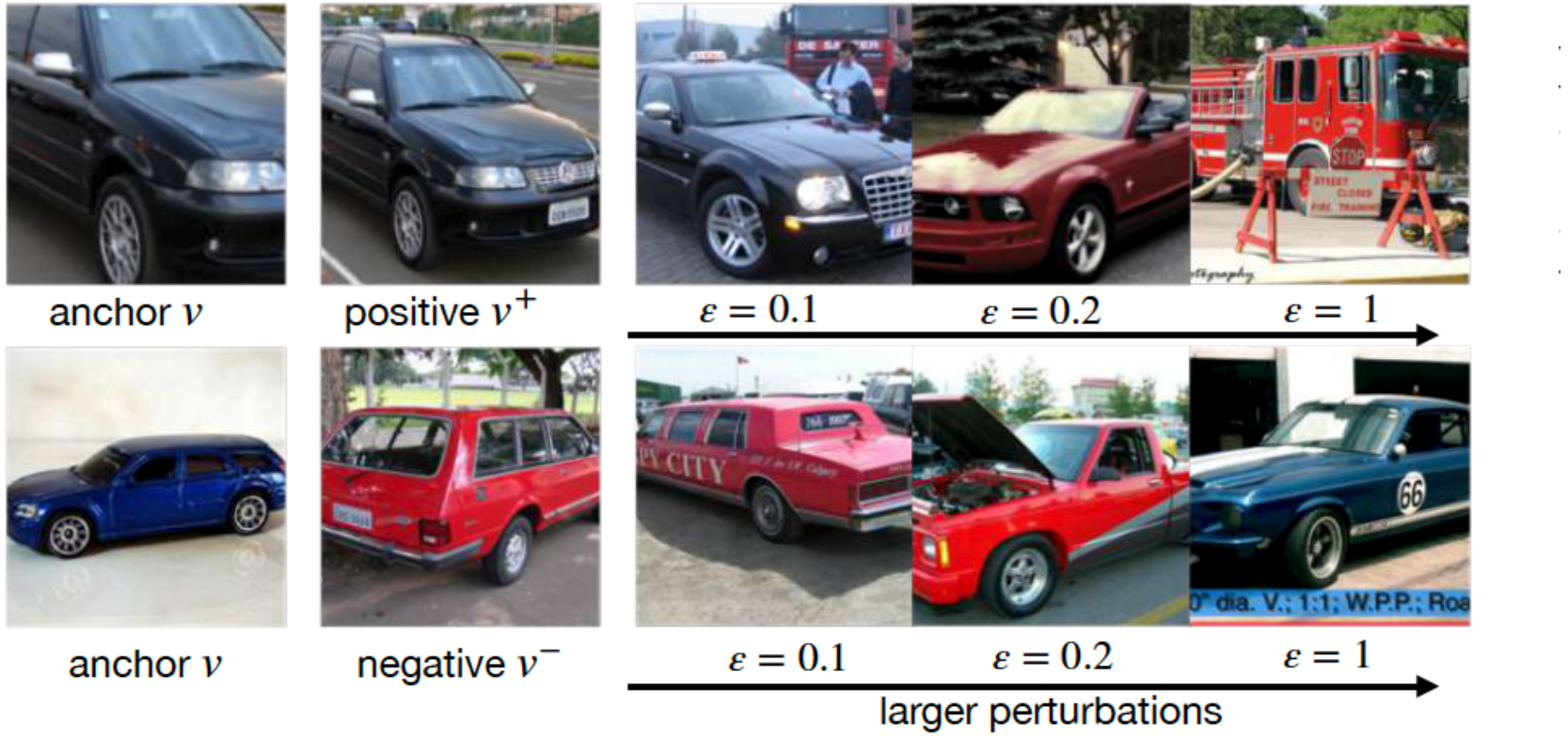

Figure 4: Visualizing implicit feature modification. **Top row:** progressively moving positive sample away from anchor. **Bottom row:** progressively moving negative sample away from anchor. In both cases, semantics such as color, orientation, and vehicle type are modified, showing the suitability of implicit feature modification for altering instance discrimination tasks.

Experimental Results

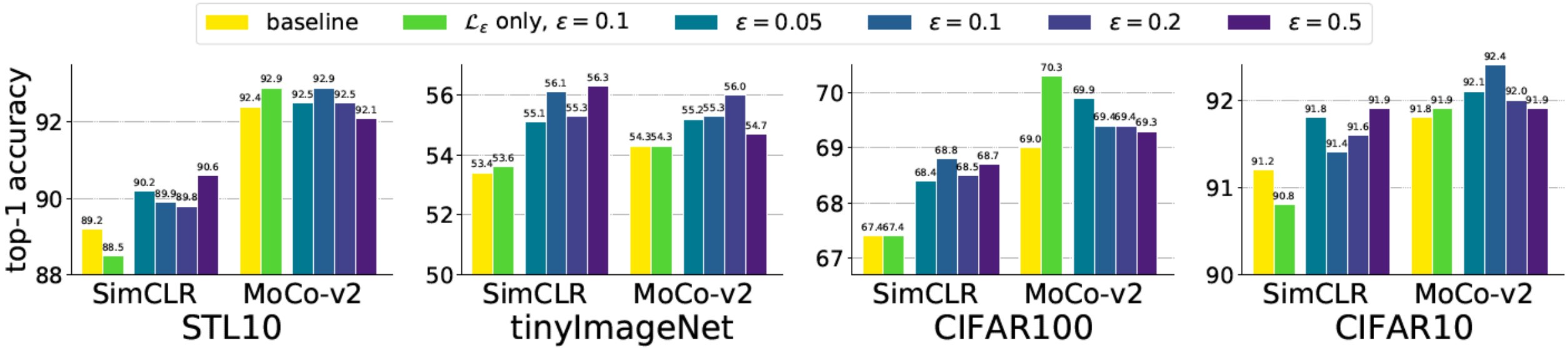

Figure 7: IFM improves linear readout performance on all datasets for all $\varepsilon \in \{0.05, 0.1, 0.2\}$ compared to SimCLR and MoCo-v2 baselines. Protocol uses 400 epochs of training with ResNet-50 backbone.

Experimental Results

Method	logFEV1pp	logFEV$_1$FVC	CLE	CLE *1-off*	Para-septal	Para-septal *1-off*	mMRC	mMRC *1-off*
Loss	R-Square		Accuracy (%)					
$\mathcal{L}$ (baseline)	$0.566_{\pm.005}$	$0.661_{\pm.005}$	$49.6_{\pm0.4}$	$81.8_{\pm0.5}$	$55.7_{\pm0.3}$	$84.4_{\pm0.2}$	$50.4_{\pm0.5}$	$72.5_{\pm0.3}$
$\mathcal{L}_\varepsilon$, $\varepsilon = 0.1$	$0.591_{\pm.008}$	$0.681_{\pm.008}$	$49.4_{\pm0.4}$	$81.9_{\pm0.3}$	$55.6_{\pm0.3}$	$85.1_{\pm0.2}$	$50.3_{\pm0.8}$	$72.7_{\pm0.4}$
IFM, $\varepsilon = 0.1$	$\mathbf{0.615_{\pm.005}}$	$\mathbf{0.691_{\pm.006}}$	$48.2_{\pm0.8}$	$80.6_{\pm0.4}$	$55.3_{\pm0.4}$	$84.7_{\pm0.3}$	$50.4_{\pm0.5}$	$72.8_{\pm0.2}$
IFM, $\varepsilon = 0.2$	$0.595_{\pm.006}$	$0.683_{\pm.006}$	$48.5_{\pm0.6}$	$80.5_{\pm0.6}$	$55.3_{\pm0.3}$	$85.1_{\pm0.1}$	$49.8_{\pm0.8}$	$72.0_{\pm0.3}$
IFM, $\varepsilon = 0.5$	$0.607_{\pm.006}$	$0.683_{\pm.005}$	$49.6_{\pm0.4}$	$82.0_{\pm0.3}$	$54.9_{\pm0.2}$	$84.7_{\pm0.2}$	$\mathbf{50.6_{\pm0.4}}$	$\mathbf{73.1_{\pm0.2}}$
IFM, $\varepsilon = 1.0$	$0.583_{\pm.005}$	$0.675_{\pm.006}$	$\mathbf{50.0_{\pm0.5}}$	$\mathbf{82.9_{\pm0.4}}$	$\mathbf{56.3_{\pm0.6}}$	$\mathbf{85.7_{\pm0.2}}$	$50.3_{\pm0.6}$	$71.9_{\pm0.3}$

Table 2: Linear readout performance on COPDGene dataset. The values are the average of 5-fold cross validation with standard deviations. IFM yields improvements on all phenotype predictions.

Experimental Results

Constructing non-robust features. Given encoder f we finetune a linear probe (classifier) h on-top of f using training data (to avoid smoothing effects we do not use data augmentation). Once h is trained, we consider each labeled example (x, y) from training data $\mathcal{D}_{\text{train}} \in \{\text{tinyImageNet, STL10, CIFAR10, CIFAR100}\}$. A hallucinated target label t is sampled uniformly at random, and we perturb $x = x_0$ until $h \circ f$ predicts t using repeated FGSM attacks [12] $x_k \leftarrow x_{k-1} - \varepsilon \text{sign}(\nabla_x \ell(h \circ f(x_{k-1}), t))$. At each step we check if $\arg\max_i h \circ f(x_k)_i = t$ (we use the maximum of logits for inference) and stop iterating and set $x_{\text{adv}} = x_k$ for the first k for which the prediction is t. This usually takes no more than a few FGSM steps with $\varepsilon = 0.01$. We form a dataset of "robust" features by adding (x_{adv}, y) to $\mathcal{D}_R$, and a dataset of "non-robust" features by adding (x_{adv}, t) to $\mathcal{D}_{NR}$. To a human the pair (x_{adv}, t) will look mislabeled, but for the encoder x_{adv} contains features predictive of t. Finally, we re-finetune (i.e. re-train) linear classifier g using $\mathcal{D}_R$ (resp. $\mathcal{D}_{NR}$) as training data.

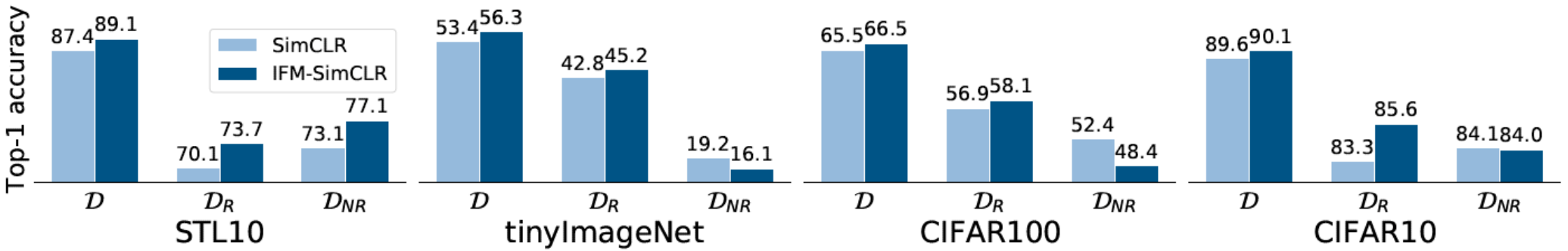

Figure 8: Label $\{\mathcal{D}, \mathcal{D}_{\text{R}}, \mathcal{D}_{\text{NR}}\}$ indicates which dataset was used to train the linear readout function. Improved performance of IFM on standard data $\mathcal{D}$ can be attributed to improved representation of *robust* features $\mathcal{D}_{\text{R}}$. See Sec. 4.3 for construction of robust ($\mathcal{D}_{\text{R}}$) and non-robust ($\mathcal{D}_{\text{NR}}$) feature datasets.


THANK YOU!


Q & A