Contrastive Learning

Representation learning via Invariant Causal Mechanisms (ReLIC)
by Mitrovic et al.

2021.09.27 – Ramith Hettiarachchi

Contents

1. Limitations in supervised learning
2. Towards self-supervised learning (SSL)
3. Contrastive learning
   - Recent progress
4. How is it performed ?
   - ReLIC Paper

Supervised Learning

- Relies on large amounts of data
- “labeled” data needed
- Generalization problems
- Adversarial Attacks

Teapot(24.99%)
Joystick(37.39%)

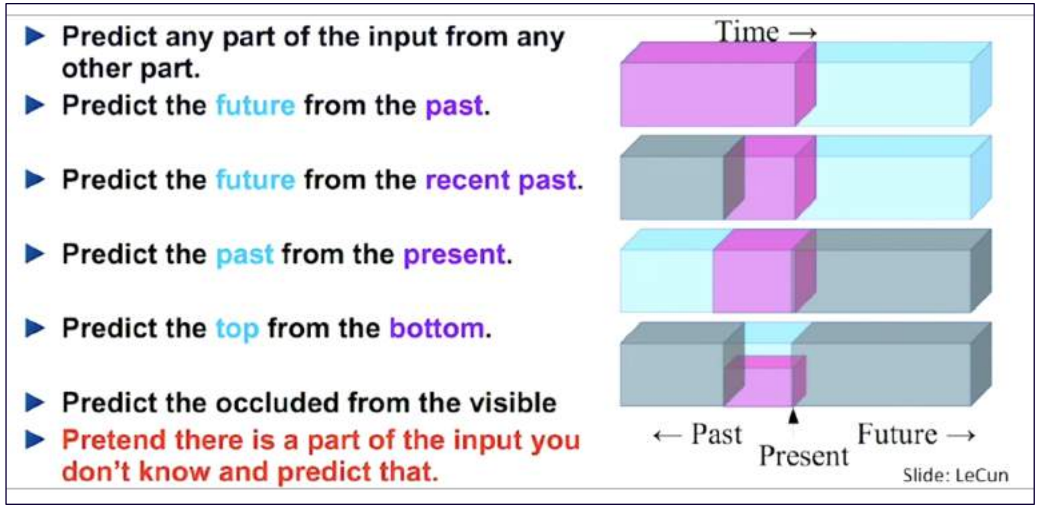
Source: https://cs.stanford.edu/people/karpathy

ImageNet – 14 Million Labelled Images

Su, Jiawei, et al. ‘One Pixel Attack for Fooling Deep Neural Networks’. *IEEE Transactions on Evolutionary Computation*,.

Towards self-supervised learning

- ✓ Orders of magnitude more data
- ✓ Get labels for free
- ✓ Idea already widely used in NLP (Eg. BERT)



Progress in terms of ,

- Non-contrastive methods (image in-painting, colorization, rotation prediction)
- Contrastive methods (MoCo, SimCLR, ReLIC..) 📈

SSL in computer vision

Generally has two steps,

- Pretext/Proxy task
  - Used to learn visual representation, with the goal of using it in the real task

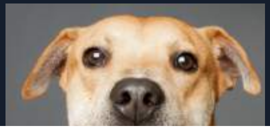
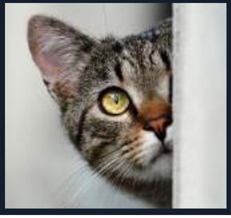
- Real (downstream) task.
  - Classification
  - Detection task
  - With insufficient annotated data samples.

Pretext Task/Proxy Task

- Non-Contrastive Approaches
    - Generate a pseudo-label $\acute{y}$ from part of the input data itself
    - Only few labels could be there
- Contrastive Approaches
    - Learn a discriminative model on multiple input pairs

Contrastive Learning




$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}, \qquad (1)$$





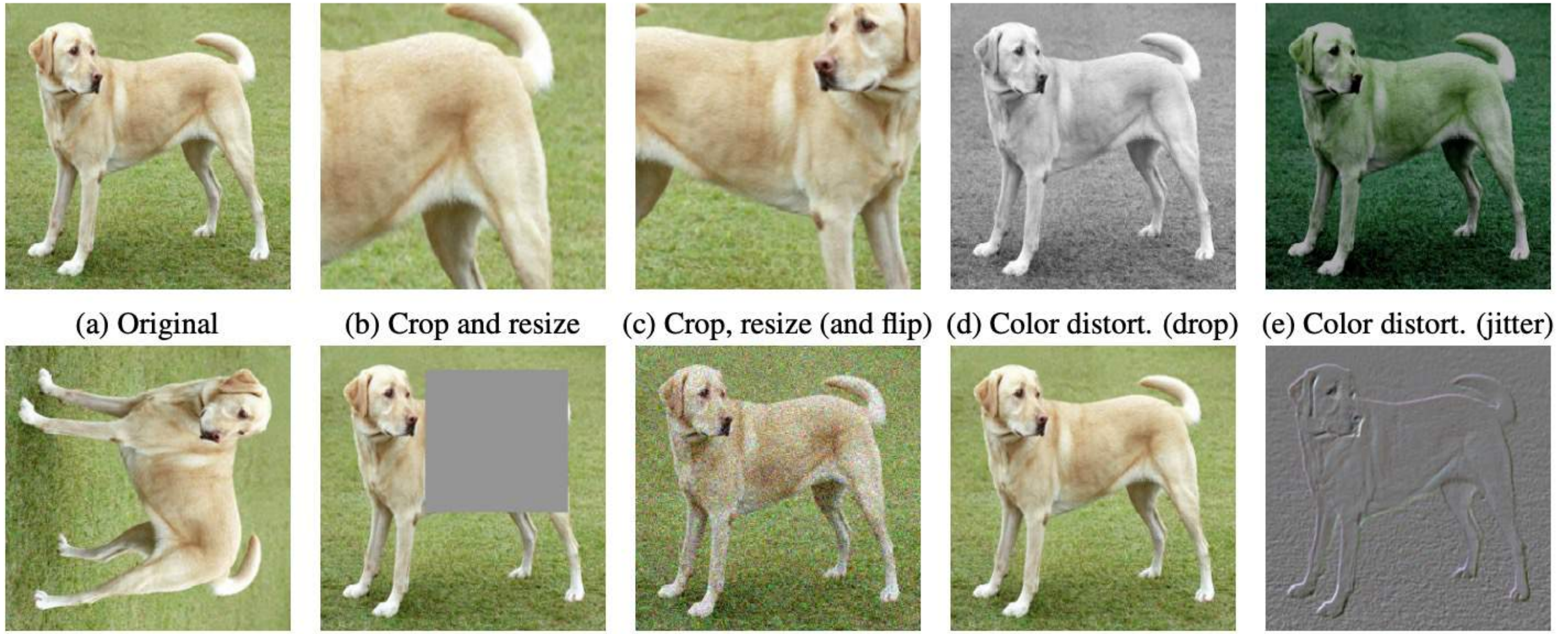

Contrastive Learning

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}, \qquad (1)$$

Chen, Ting, et al. 'A Simple Framework for Contrastive Learning of Visual Representations'. *ArXiv:2002.05709 [Cs, Stat]*, June 2020. *arXiv.org*, http://arxiv.org/abs/2002.05709.

Pretext Task/Proxy Task

Pretext Task/Proxy Task



(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

(f) Rotate {90°, 180°, 270°} (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

Chen, Ting, et al. ‘A Simple Framework for Contrastive Learning of Visual Representations’. *ArXiv:2002.05709 [Cs, Stat]*, June 2020. *arXiv.org*, http://arxiv.org/abs/2002.05709.

Progress in self-supervised learning

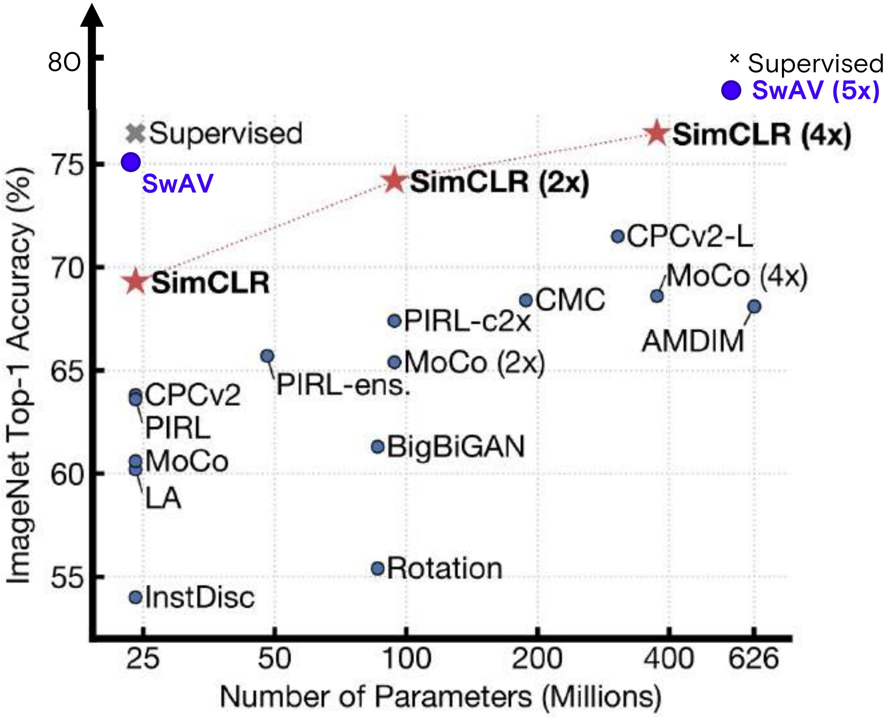


Figure 1. ImageNet Top-1 accuracy of linear classifiers trained on representations learned with different self-supervised methods (pretrained on ImageNet). Gray cross indicates supervised ResNet-50. Our method, SimCLR, is shown in bold.

SimCLR
2020 – Feb

Chen, Ting, et al. 'A Simple Framework for Contrastive Learning of Visual Representations'. *ArXiv:2002.05709 [Cs, Stat]*, June 2020.

Representation learning via Invariant Causal Mechanisms (ReLIC)

Authors: Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, Charles Blundell

Motivating Problem

💭 How to learn useful representations, when we don't have access to labels?

Approach :

First understand what needs to be learnt, then how to learnt it

Notations

X – Unlabeled data

$$\mathcal{Y} = \{Y_t\}_{\{t=1\}}^{T}$$

Set of downstream tasks

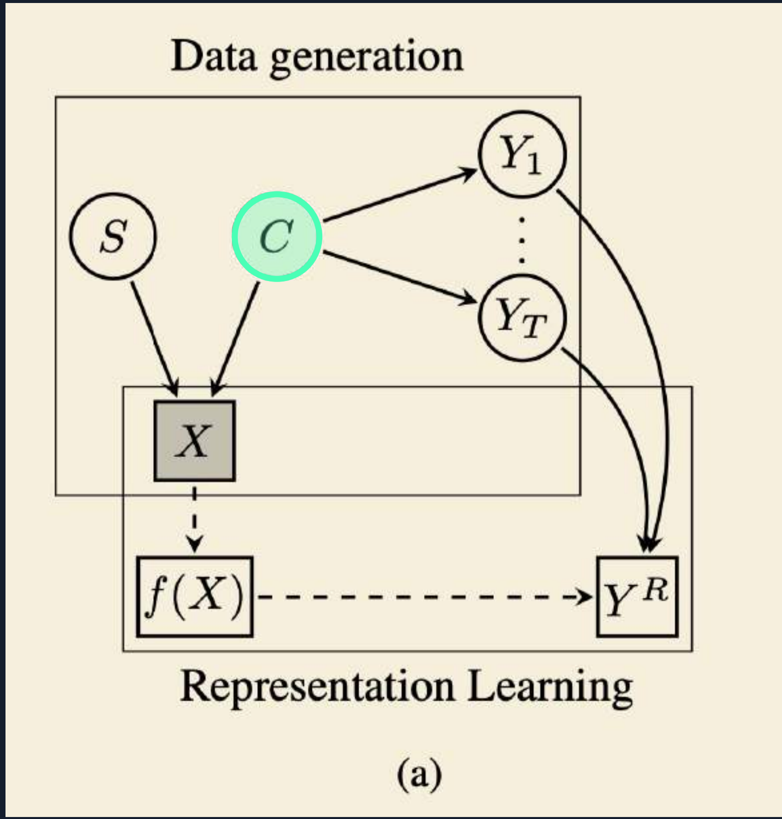
targets for task t

Assumptions on data generation

1. Data (X) = Style (S) + Content (C)
2. Content is only what matters for downstream tasks
3. S and C are independent

What needs to be learnt?

$\longrightarrow$ Content 💡



Formalize,

Performing interventions on S does not change the conditional distribution $P(Y_t|C)$

$$p^{do(S=s_i)}(Y_t \mid C) = p^{do(S=s_j)}(Y_t \mid C) \qquad \forall\, s_i, s_j \in S$$

$p^{do(S=s)}$ ← denotes the distribution arising from assigning S the value s

Few things to think about ..

- The targets of the downstream tasks (Y_t) are unknown
- We don't have access to S

- Construct a proxy task Y^R
- Use content preserving style augmentations – a_i

  (rotations, grayscaling, translation, cropping)

- Construct a proxy task Y^R
- Use content preserving style augmentations – a_i

(rotations, grayscaling, translation, cropping)

Towards ReLIC Objective,

$$p^{do(S=s_i)}(Y_t \mid C) = p^{do(S=s_j)}(Y_t \mid C) \qquad \forall\, s_i, s_j \in S$$

- Construct a proxy task Y^R
- Use content preserving style augmentations – a_i

  (rotations, grayscaling, translation, cropping)

Towards ReLIC Objective,

$$p^{do(S=s_i)}(Y_t \mid \qquad) = p^{do(S=s_j)}(Y_t \mid \qquad) \qquad \forall\, s_i, s_j \in S$$

- Construct a proxy task Y^R
- Use content preserving style augmentations - a_i

  (rotations, grayscaling, translation, cropping)

Towards ReLIC Objective,

$$p^{do(S=s_i)}(Y_t \mid f(x)) = p^{do(S=s_j)}(Y_t \mid f(x)) \qquad \forall\, s_i, s_j \in S$$

- Construct a proxy task Y^R
- Use content preserving style augmentations – a_i

  (rotations, grayscaling, translation, cropping)

Towards ReLIC Objective,

$$p^{do(S=s_i)}(Y^R \mid f(x)) = p^{do(S=s_j)}(Y^R \mid f(x)) \qquad \forall\, s_i, s_j \in S$$

- Construct a proxy task Y^R
- Use content preserving style augmentations - a_i

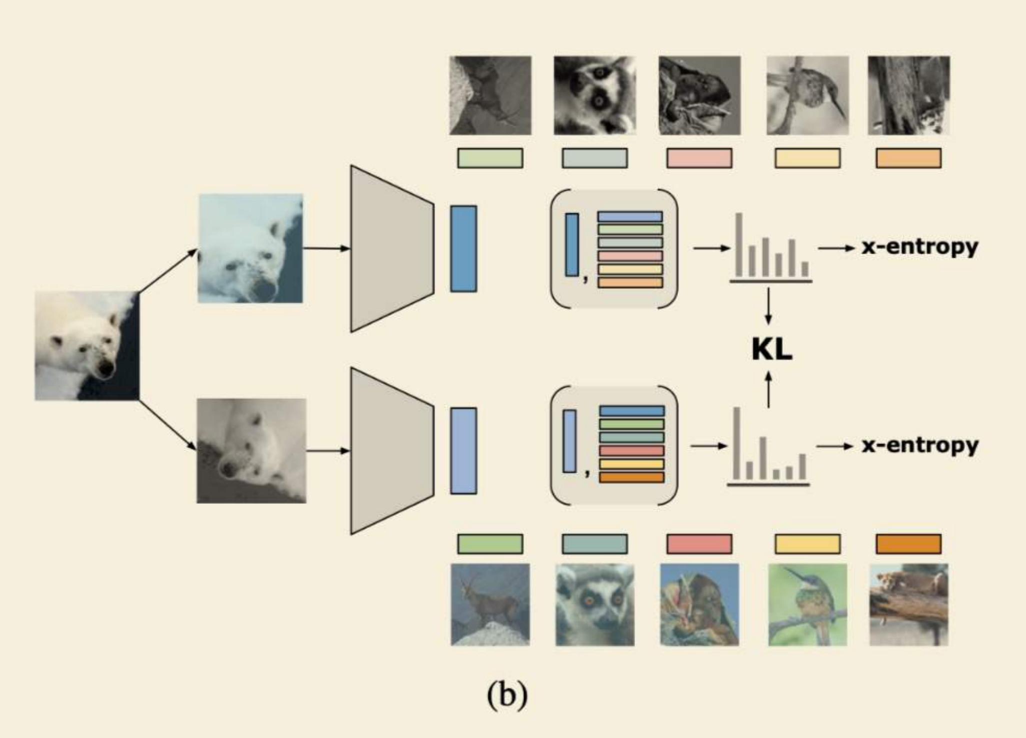
  (rotations, grayscaling, translation, cropping)

Towards ReLIC Objective,

$$p^{do(S=a_i)}(Y^R \mid f(x)) = p^{do(S=a_j)}(Y^R \mid f(x)) \qquad \forall\, a_i, a_j \in S$$

ReLIC Objective,

$$p^{do(S=a_i)}(Y^R |\, f(x)\,) = p^{do(S=a_j)}(Y^R |\, f(x)) \qquad \forall\, a_i, a_j\ \in S$$



Enforcing the ReLIC Objective as a **regularizer**

$$p^{do(S=a_i)}(Y^R | f(x)) = p^{do(S=a_j)}(Y^R | f(x)) \qquad \forall\, a_i, a_j \in S$$

Minimize the following objective, over $x_i \in \mathcal{D}$,

$$-\sum_{i=1}^{N} \sum_{a_{lk}} \log \frac{\exp\left(\frac{\phi\left(f\left(x_i^{a_l}\right), h\left(x_i^{a_k}\right)\right)}{\tau}\right)}{\sum_{m=1}^{M} \exp\left(\frac{\phi\left(f\left(x_i^{a_l}\right), h\left(x_i^{a_k}\right)\right)}{\tau}\right)} + \alpha \sum_{a_{lk}, a_{qt}} KL\left(p^{do(S=a_{lk})}, p^{do(S=a_{qt})}\right)$$

- $a_{lk} \in \mathcal{A} \times \mathcal{A}$
- α the weighting of the invariance penalty
- M the number of points we use to construct the contrast set

$p^{do(S=a_{lk})}(Y_R = j \mid f(x_i)),$

Enforcing the ReLIC Objective as a regularizer

$$p^{do(S=a_i)}(Y^R \mid f(x)) = p^{do(S=a_j)}(Y^R \mid f(x)) \qquad \forall\, a_i, a_j \in S$$

f

h

h

Minimize the following objective, over $x_i \in \mathcal{D}$,

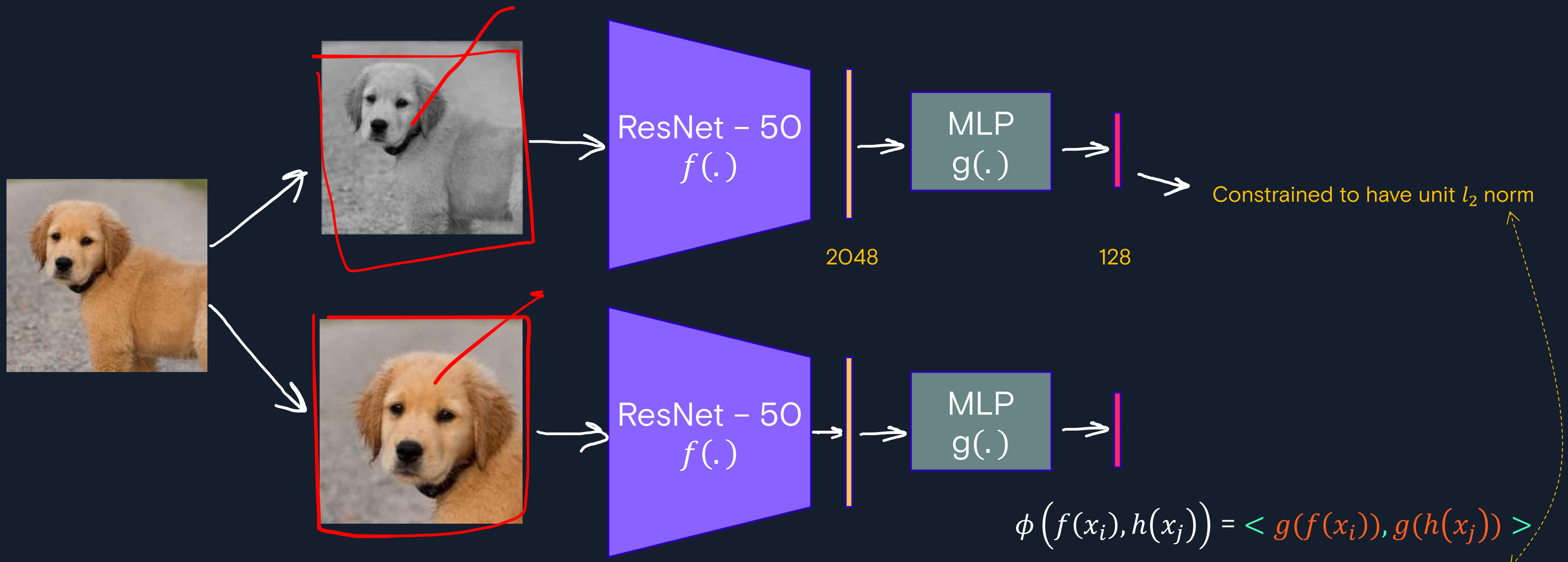
t

$$- \sum_{i=1}^{N} \sum_{a_{lk}} \log \frac{\exp\left(\frac{\phi\left(f\left(x_i^{a_l}\right), h\left(x_i^{a_k}\right)\right)}{\tau}\right)}{\sum_{m=1}^{M} \exp\left(\frac{\phi\left(f\left(x_i^{a_l}\right), h\left(x_i^{a_k}\right)\right)}{\tau}\right)} + \alpha \sum_{a_{lk}, a_{qt}} KL\left(p^{do(S=a_{lk})}, p^{do(S=a_{qt})}\right)$$

- $a_{lk} \in \mathcal{A} \times \mathcal{A}$
- α the weighting of the invariance penalty
- M the number of points we use to construct the contrast set

$$p^{do(S=a_{lk})}(Y_R = j \mid f(x_i)),$$

$$-\sum_{i=1}^{N}\sum_{a_{lk}}\log\frac{\exp\left(\frac{\phi\left(f\left(x_i^{a_l}\right),h\left(x_i^{a_k}\right)\right)}{\tau}\right)}{\sum_{m=1}^{M}\exp\left(\frac{\phi\left(f\left(x_i^{a_l}\right),h\left(x_i^{a_k}\right)\right)}{\tau}\right)} + \alpha\sum_{a_{lk},a_{qt}} KL\left(p^{do(S=a_{lk})}, p^{do(S=a_{qt})}\right)$$



$$\phi\left(f(x_i), h(x_j)\right) = \langle g(f(x_i)), g(h(x_j)) \rangle$$

(Dot product, → Becomes cosine Similarity)

Representation learning via Invariant Causal Mechanisms by Mitrovic et al.

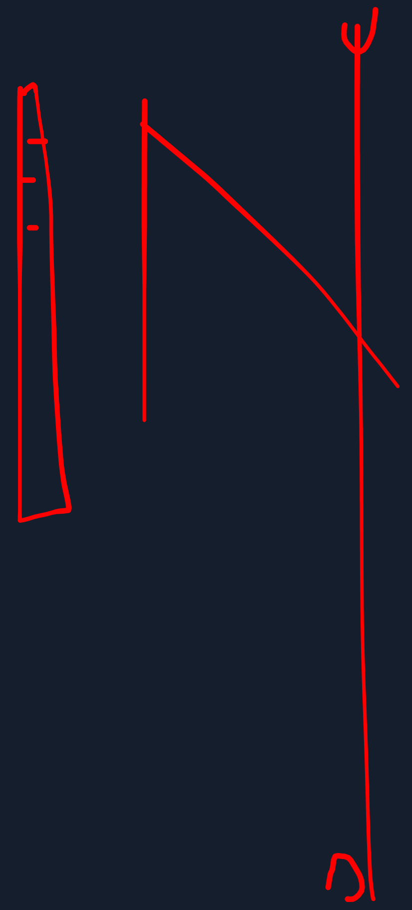
Linear Evaluation on Image Net

Table 1: Accuracy (in %) under linear evaluation on ImageNet for different self-supervised representation learning methods. Methods with * use SimCLR augmentations. Methods with † use custom, stronger augmentations.

Method		Top-1	Top-5
ResNet-50 architecture			
PIRL		63.6	-
CPC v2		63.8	85.3
CMC		66.2	87.0
SimCLR [4]	*	69.3	89.0
SwAV [2]	*	70.1	-
RELIC (ours)	*	70.3	89.5
InfoMin Aug. [22]	†	73.0	91.1
SwAV [2]	†	75.3	-
ResNet-50 with target network			
MoCo v2 [5]		71.1	-
BYOL [7]	*	74.3	91.6
RELIC (ours)	*	74.8	92.2

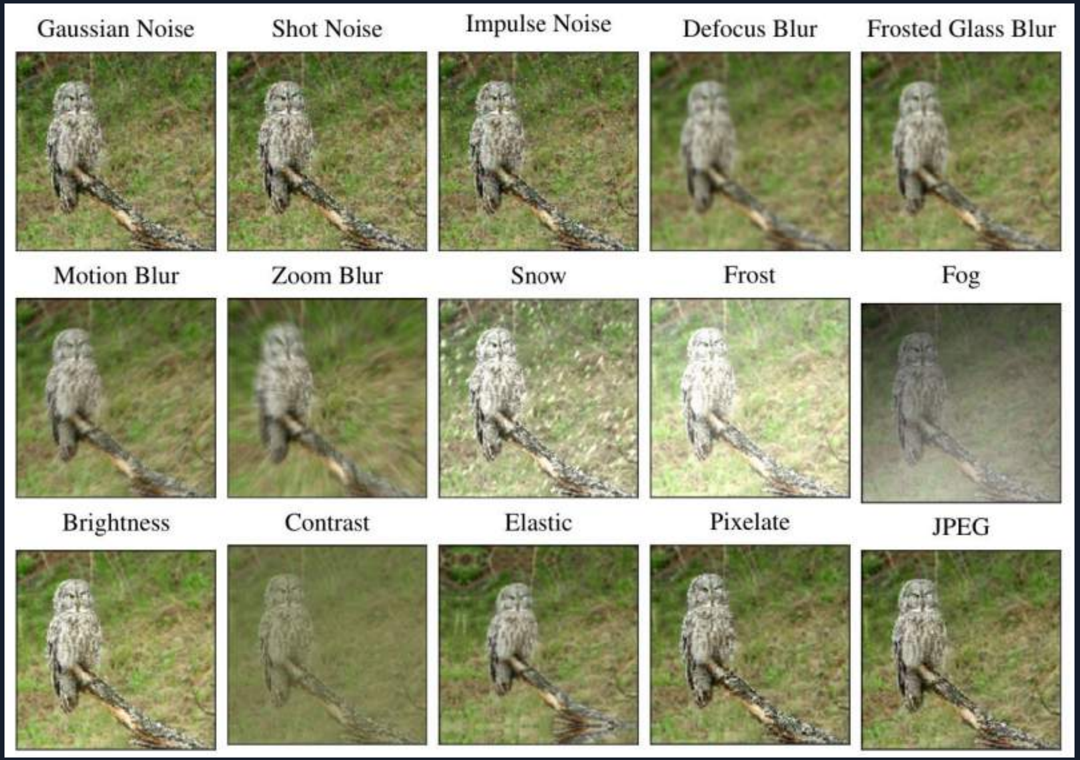
* uses standard augmentations

† uses stronger augmentations



Representation learning via Invariant Causal Mechanisms by Mitrovic et al.

Robustness



ImageNet-C (corrupted)
Images with diverse corruptions of varying strengths
Tests: robustness of representation

Method	Supervised	SimCLR	ReLIC	BYOL	ReLIC_T
mCE (%)	76.7	87.5	76.4	72.3	70.8

Out-of-distribution generalization

ImageNet-R (rendered)
New renditions of 200 ImageNet classes
Tests: out-of-distribution generalization

Method	Supervised	SimCLR	ReLIC	BYOL	ReLIC$_T$
Top-1 Error (%)	63.9	81.7	77.4	77.0	76.2

Summary

- Formalize problem of self-supervised representation learning using causality and propose to more effectively leverage data augmentations through invariant prediction.

- New self-supervised objective, REpresentation Learning with Invariance Causal mechanisms (RELIC), that enforces invariant prediction through an explicit regularizer and show improved generalization guarantees.

- Generalize contrastive learning using refinements and show that learning on refinements is a sufficient condition for learning useful representations; this provides an alternative explanation to MI for the success of contrastive methods

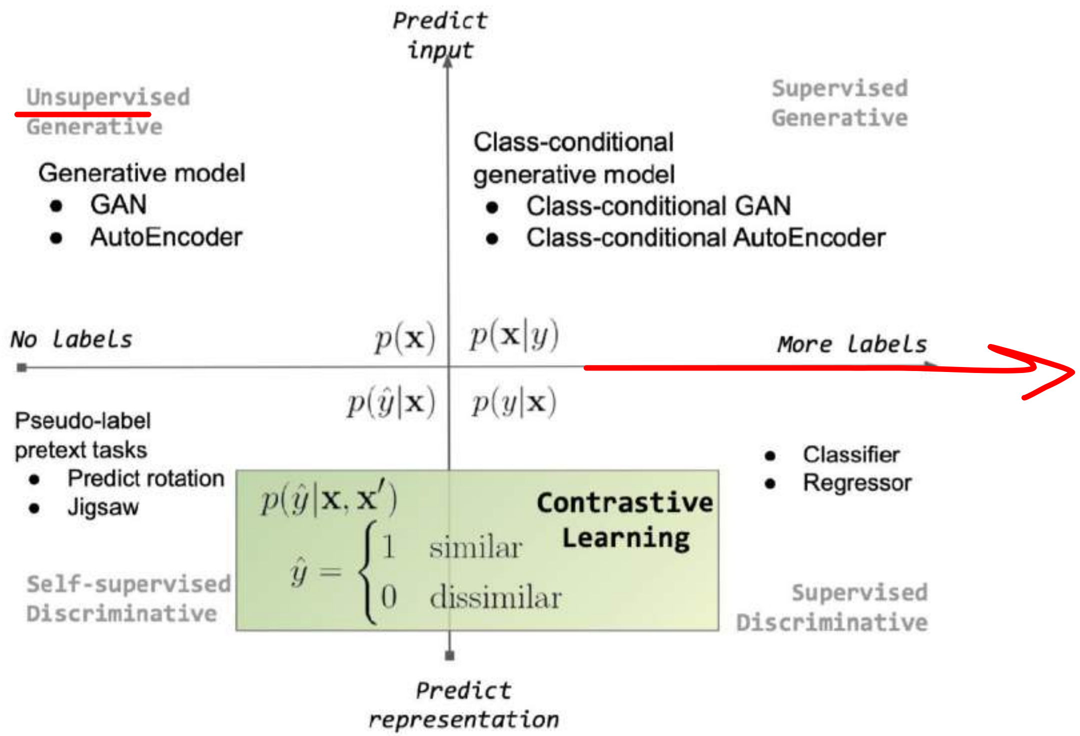


FIGURE 1. **Contrastive learning in the Generative-Discriminative and Supervised-Unsupervised spectrum. Contrastive methods belong to the group of discriminative models that predict a pseudo-label of *similarity* or *dissimilarity* given a pair of inputs.**

Figure 1 illustrates the family of contrastive methods along generative-discriminative and supervised-unsupervised axes.

Useful Reading

- Self-Supervised Representation Learning (lilianweng.github.io)
- Self-supervised learning: The dark matter of intelligence (facebook.com)
- Le-Khac, Phuc H., et al. 'Contrastive Representation Learning: A Framework and Review'. *IEEE Access*, vol. 8, 2020, pp. 193907–34.

Architecture

f(x)

g(x)

E.2 ARCHITECTURE

We test RELIC on two different architectures – ResNet-50 (He et al., 2016) and ResNet-50 with target network as in (Grill et al., 2020). For ResNet-50, we use version 1 with post-activation. We take the representation to be the output of the final average pooling layer, which is of dimension 2048. As in SimCLR (Chen et al., 2020a), we use a critic network to project the representation to a lower dimensional space with a multi-layer perceptron (MLP). When using ResNet-50 as encoder, we treat the parameters of the MLP (e.g. depth and width) as hyperparameters and sweep over them. This MLP has batch normalization (Ioffe & Szegedy, 2015) after every layer, rectified linear activations (ReLU) (Nair & Hinton, 2010). We used a 4 layer MLP with widths $[4096, 2048, 1024, 512]$ and output size 128 with ResNet-50. When using a ResNet-50 with target networks as in (Grill et al., 2020), we exactly follow their architecture settings.