Contrastive Learning for Novelty Detection

Jinwoo Shin

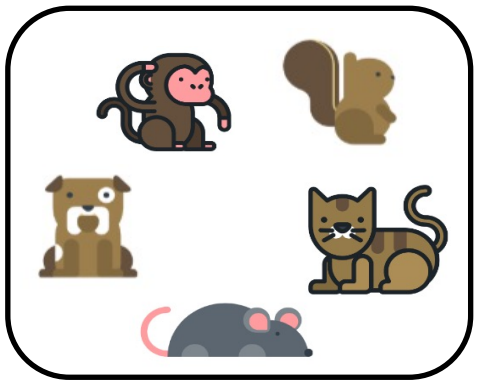
Korea Advanced Institute of Science and Technology (KAIST)

Joint work with Jihoon Tack* (KAIST), Sangwoo Mo* (KAIST), Jongheon Jeong (KAIST)

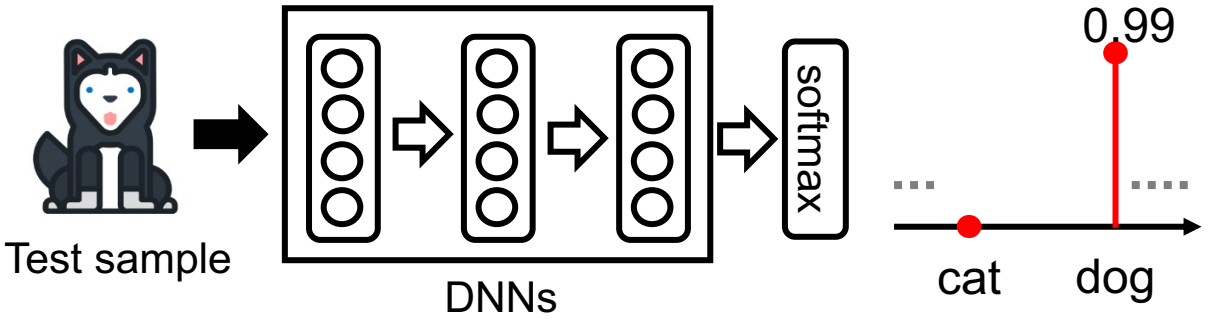
Paper draft is available at NeurIPS 2020 and https://arxiv.org/abs/2007.08176

Novelty/Out-of-distribution Detection

Deep neural networks (DNNs) generalize well under the "seen" test distribution



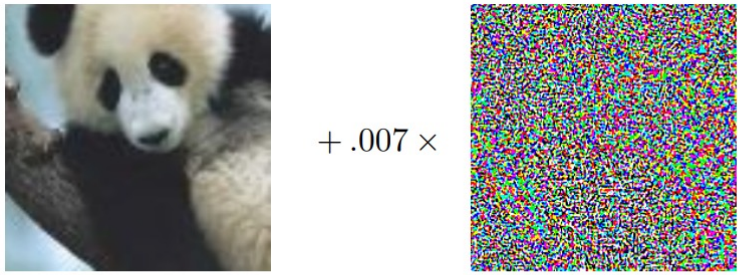
Training data: "Animals"



For "unseen" distributions, DNNs often show astonishingly unexpected behaviors

Out-of-distribution

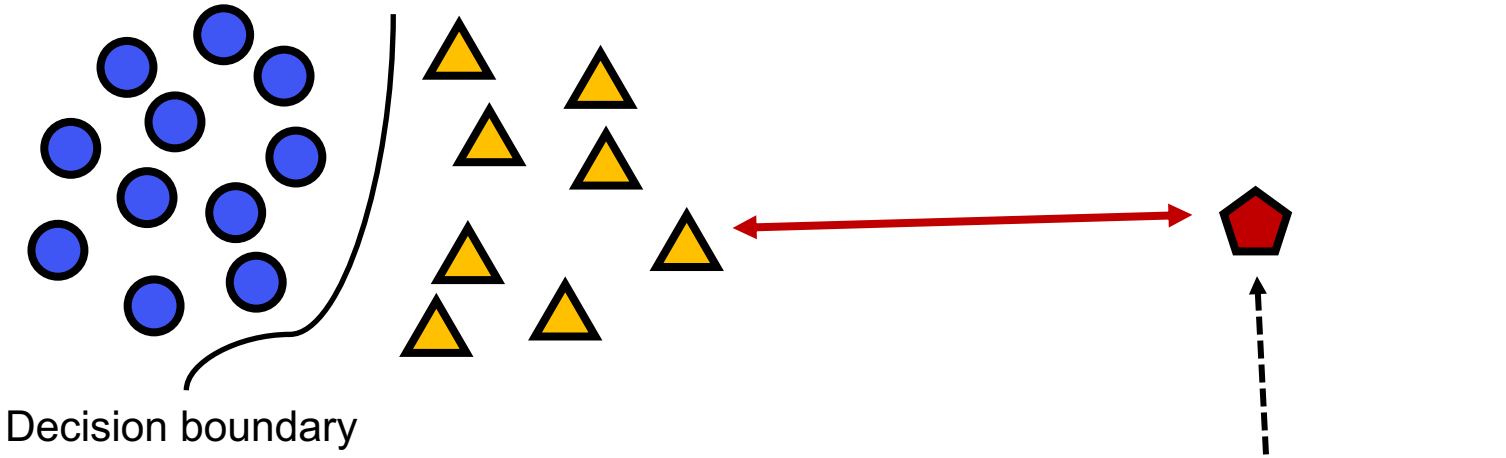
Unidentifiable



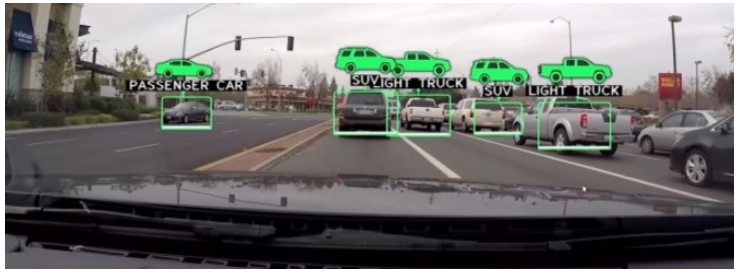
Adversarial samples

Novelty/Out-of-distribution Detection

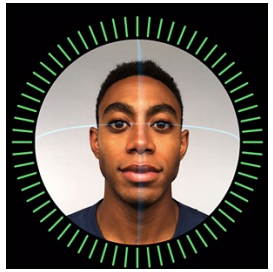
Is it possible to figure out whether a given sample is out-of-distribution (OOD)?



Practically, such an ability is indispensable for security-concerned systems



Autonomous drive



Secure authentication system

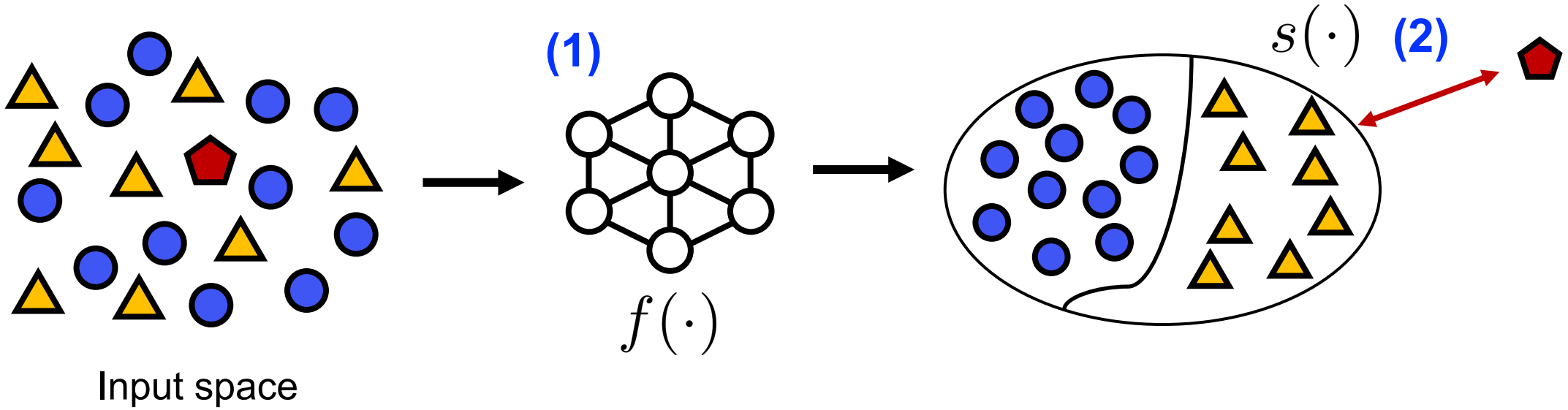
Novelty/Out-of-distribution Detection

Is it possible to figure out whether a given sample is out-of-distribution (OOD)?

1. How to learn a better representation $f(\cdot)$ more suitable for OOD detection?
2. How to define a detection score $s(\cdot)$ that maximally utilizes $f(\cdot)$?

$$s(\bigcirc) > s(\pentagon)$$

: Out-of-distribution

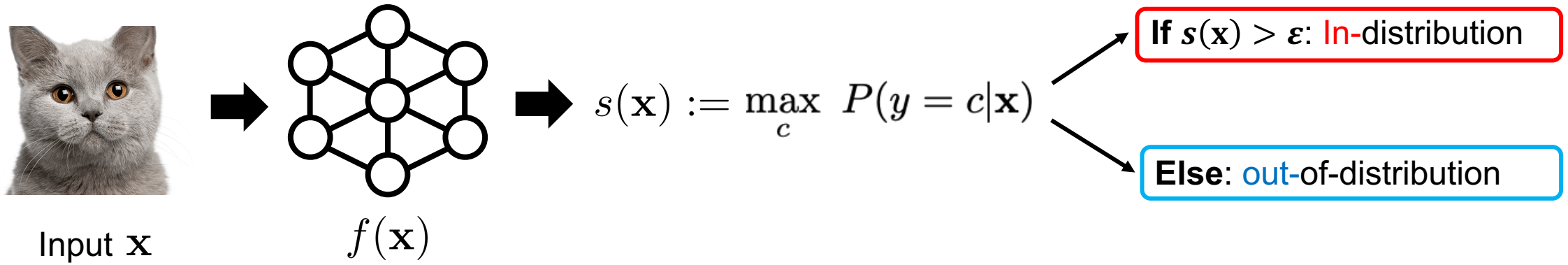


Novelty/Out-of-distribution Detection

Case 1: $f(\cdot) =$ a pre-trained classifier from a labeled dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$

1. ~~How to learn a better representation $f(\cdot)$ more suitable for OOD detection?~~
2. How to define a detection score $s(\cdot)$ that maximally utilizes $f(\cdot)$?

- **Example 1** (the "Baseline" detector)**:** Max-confidence based score [Hendrycks et al., 2017]



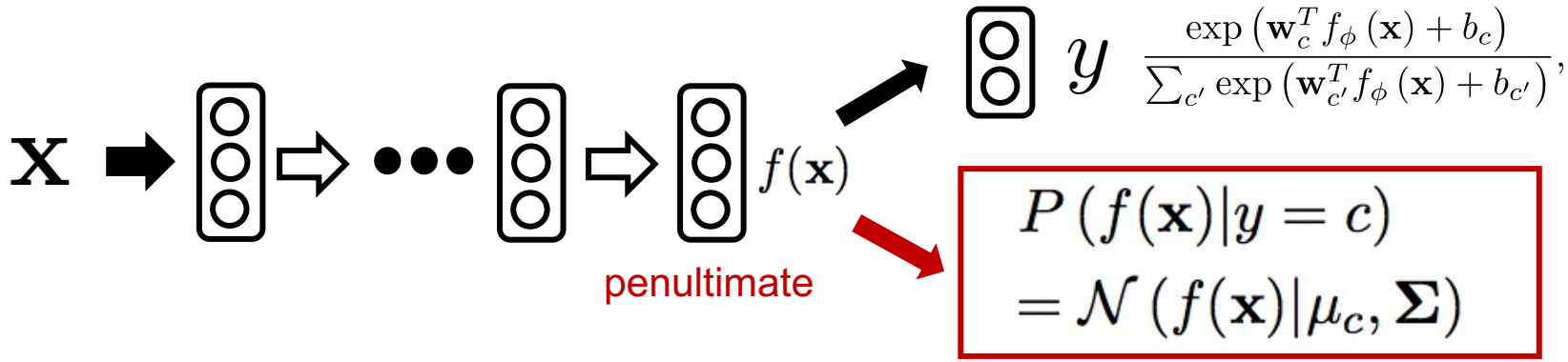
Hendrycks & Gimpel. A Baseline for Detecting Misclassified and Out-of-distribution Examples in Neural Networks. ICLR 2017.

Novelty/Out-of-distribution Detection

Case 1: $f(\cdot)$ = a pre-trained classifier from a labeled dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$

1. ~~How to learn a better representation $f(\cdot)$ more suitable for OOD detection?~~
2. How to define a detection score $s(\cdot)$ that maximally utilizes $f(\cdot)$?

- **Example 2:** Mahalanobis-based confidence score [Lee et al., 2018]
  - Define a generative classifier $P(\mathbf{x}|y)$ from intermediate features



$$\widehat{\mu}_c = \frac{1}{N_c} \sum_{i:y_i=c} f(\mathbf{x}_i), \quad \widehat{\mathbf{\Sigma}} = \frac{1}{N} \sum_c \sum_{i:y_i=c} (f(\mathbf{x}_i) - \widehat{\mu}_c)(f(\mathbf{x}_i) - \widehat{\mu}_c)^\top,$$

Lee, Lee, Lee & Shin. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks, NeurIPS 2018.

Novelty/Out-of-distribution Detection

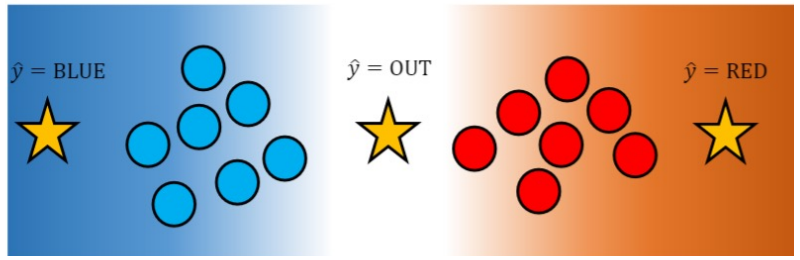
Case 1: $f(\cdot) =$ a pre-trained classifier from a labeled dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$

1. ~~How to learn a better representation $f(\cdot)$ more suitable for OOD detection?~~
2. How to define a detection score $s(\cdot)$ that maximally utilizes $f(\cdot)$?

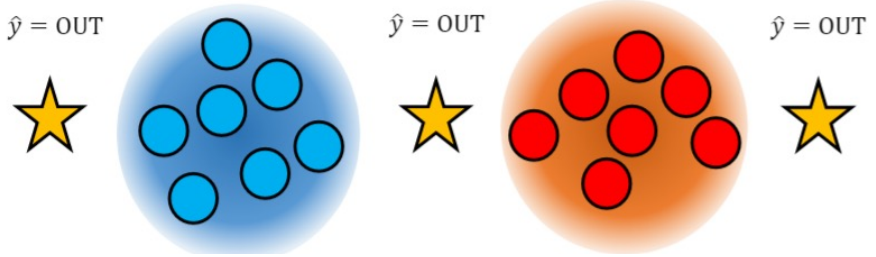
- **Example 2:** Mahalanobis-based confidence score [Lee et al., 2018]
  - Define a generative classifier $P(\mathbf{x}|y)$ from intermediate features

$$s(\mathbf{x}) := \max_c \; -(f(\mathbf{x}) - \hat{\mu}_c)^\top \hat{\mathbf{\Sigma}}^{-1} (f(\mathbf{x}) - \hat{\mu}_c)$$

$P(y|\mathbf{x})$: Discriminative, confidence-based



$P(\mathbf{x}|y)$: Generative, Mahalanobis-based



Lee, Lee, Lee & Shin. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks, NeurIPS 2018.

Novelty/Out-of-distribution Detection

Case 1: $f(\cdot)$ = a pre-trained classifier from a labeled dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$

1. ~~How to learn a better representation $f(\cdot)$ more suitable for OOD detection?~~
2. How to define a detection score $s(\cdot)$ that maximally utilizes $f(\cdot)$?

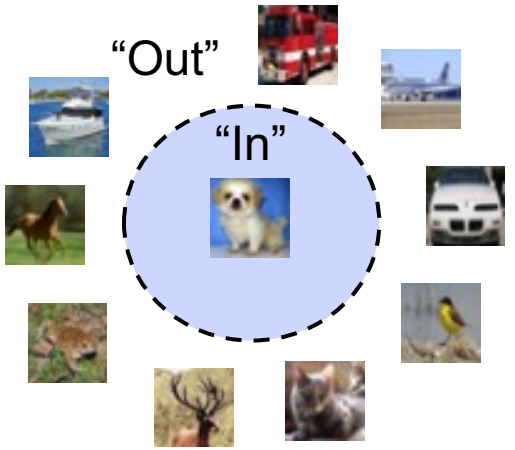
- **Example 2:** Mahalanobis-based confidence score [Lee et al., 2018]

  (+) Near-perfect detection performances for "easy"-OODs, e.g., CIFAR-10 vs. LSUN

  (–) Still challenging on "harder"-OODs: e.g., CIFAR-10 vs. CIFAR-100 / One-class CIFAR-10

In-dist (model)	OOD	TNR at TPR 95%	AUROC	Detection Acc.
		Baseline / ODIN / Mahalanobis / Ours		
CIFAR-10 (ResNet)	iSUN	44.6 / 73.2 / 97.8 / **99.3**	91.0 / 94.0 / 99.5 / **99.8**	85.0 / 86.5 / 96.7 / **98.1**
	LSUN (R)	49.8 / 82.1 / 98.8 / **99.6**	91.0 / 94.1 / 99.7 / **99.9**	85.3 / 86.7 / 97.7 / **98.6**
	LSUN (C)	48.6 / 62.0 / 81.3 / **89.8**	91.9 / 91.2 / 96.7 / **97.8**	86.3 / 82.4 / 90.5 / **92.6**
	TinyImgNet (R)	41.0 / 67.9 / 97.1 / **98.7**	91.0 / 94.0 / 99.5 / **99.7**	85.1 / 86.5 / 96.3 / **97.8**
	TinyImgNet (C)	46.4 / 68.7 / 92.0 / **96.7**	91.4 / 93.1 / 98.6 / **99.2**	85.4 / 85.2 / 93.9 / **96.1**
	SVHN	50.5 / 70.3 / 87.8 / **97.6**	89.9 / 96.7 / 99.1 / **99.5**	85.1 / 91.1 / 95.8 / **96.7**
	CIFAR-100	33.3 / **42.0** / 41.6 / 32.9	86.4 / 85.8 / **88.2** / 79.0	80.4 / 78.6 / **81.2** / 71.7

Results from [Sastry et al., 2020]



Lee, Lee, Lee & Shin. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks, NeurIPS 2018.
Sastry and Oore. Detecting Out-of-Distribution Examples with Gram Matrices, ICML 2020.

Novelty/Out-of-distribution Detection

Case 1: $f(\cdot) =$ a pre-trained classifier from a labeled dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$

1. How to learn a better representation $f(\cdot)$ more suitable for OOD detection?
2. How to define a detection score $s(\cdot)$ that maximally utilizes $f(\cdot)$?

- Ex
  (+)
  (–) Still challenging on "harder" -OODs: e.g., CIFAR-10 vs. CIFAR-100 / One-class CIFAR-10

Supervised representations can discriminate easy-OODs, but may not be enough for hard-OODs

In-dist (model)	OOD	TNR at TPR 95%	AUROC	Detection Acc.
		Baseline / ODIN / Mahalanobis / Ours		
CIFAR-10 (ResNet)	iSUN	44.6 / 73.2 / 97.8 / **99.3**	91.0 / 94.0 / 99.5 / **99.8**	85.0 / 86.5 / 96.7 / **98.1**
	LSUN (R)	49.8 / 82.1 / 98.8 / **99.6**	91.0 / 94.1 / 99.7 / **99.9**	85.3 / 86.7 / 97.7 / **98.6**
	LSUN (C)	48.6 / 62.0 / 81.3 / **89.8**	91.9 / 91.2 / 96.7 / **97.8**	86.3 / 82.4 / 90.5 / **92.6**
	TinyImgNet (R)	41.0 / 67.9 / 97.1 / **98.7**	91.0 / 94.0 / 99.5 / **99.7**	85.1 / 86.5 / 96.3 / **97.8**
	TinyImgNet (C)	46.4 / 68.7 / 92.0 / **96.7**	91.4 / 93.1 / 98.6 / **99.2**	85.4 / 85.2 / 93.9 / **96.1**
	SVHN	50.5 / 70.3 / 87.8 / **97.6**	89.9 / 96.7 / 99.1 / **99.5**	85.1 / 91.1 / 95.8 / **96.7**
	CIFAR-100	33.3 / **42.0** / 41.6 / 32.9	86.4 / 85.8 / **88.2** / 79.0	80.4 / 78.6 / **81.2** / 71.7

Results from [Sastry et al., 2020]

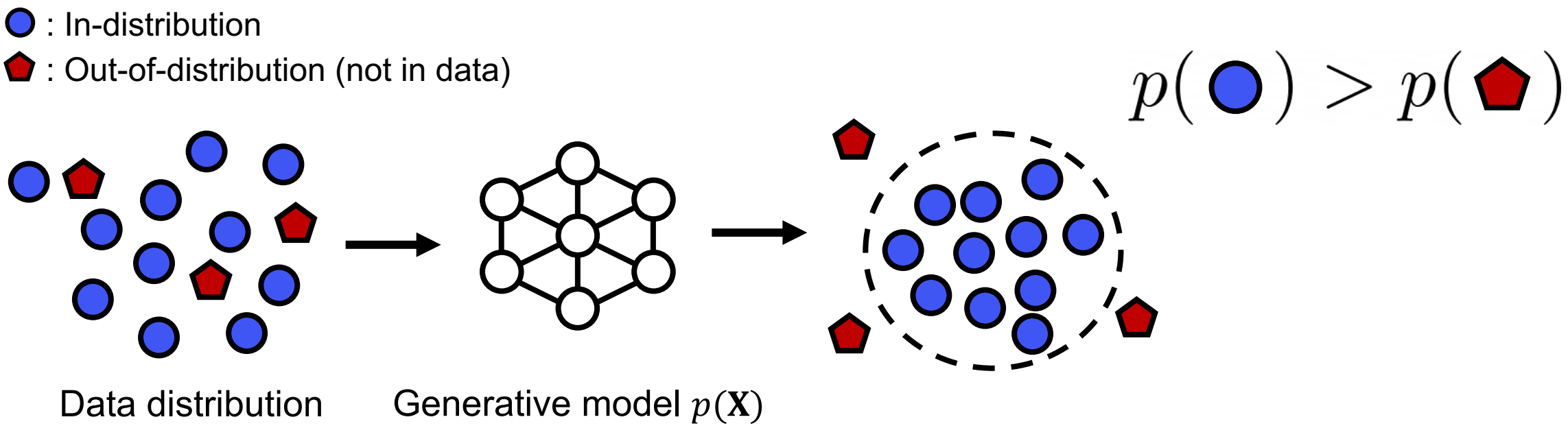
Lee, Lee, Lee & Shin. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks, NeurIPS 2018.
Sastry and Oore. Detecting Out-of-Distribution Examples with Gram Matrices, ICML 2020.

Novelty/Out-of-distribution Detection

Case 2: $f(\cdot)$ = a generative model from an unlabeled dataset $\mathcal{D} = \{\mathbf{x}_i\}$

1. How to learn a better representation $f(\cdot)$ more suitable for OOD detection?
2. ~~How to define a detection score $s(\cdot)$ that maximally utilizes $f(\cdot)$?~~

Ideally, a good likelihood model $p(\mathbf{X})$ should also represent a good $s(\cdot)$



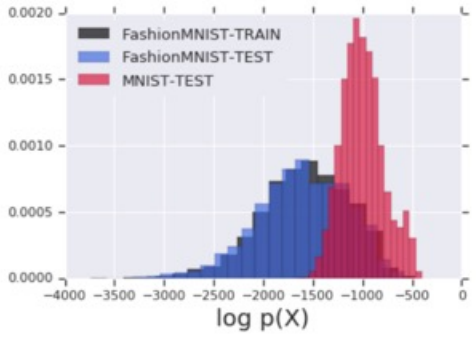
Novelty/Out-of-distribution Detection

Case 2: $f(\cdot)$ = a generative model from an unlabeled dataset $\mathcal{D} = \{\mathbf{x}_i\}$

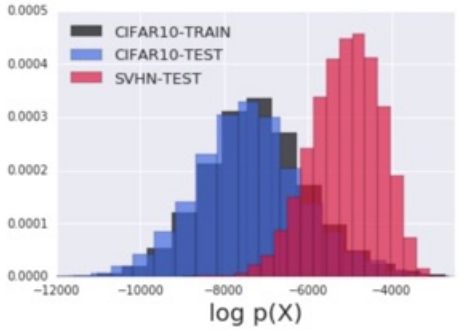
1. How to learn a better representation $f(\cdot)$ more suitable for OOD detection?
2. ~~How to define a detection score $s(\cdot)$ that maximally utilizes $f(\cdot)$?~~

Ideally, a good likelihood model $p(\mathbf{X})$ should also represent a good $s(\cdot)$

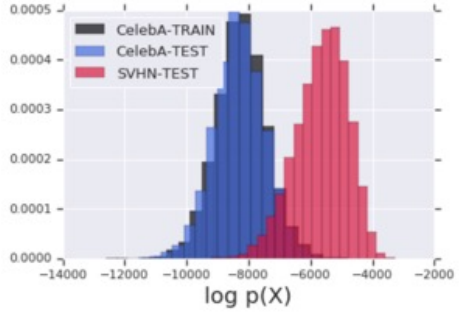
- Unfortunately, this is not the case at least for the current models
  1. They tend to be easily biased, e.g., to background statistics [Ren et al., 2019]
  2. Scaling up for a better likelihood model is usually much more challenging



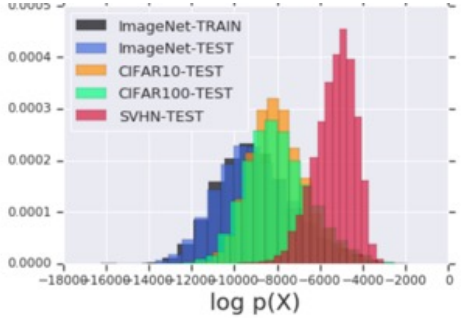
(a) Train on FashionMNIST, Test on MNIST



(b) Train on CIFAR-10, Test on SVHN



(c) Train on CelebA, Test on SVHN



(d) Train on ImageNet,
Test on CIFAR-10 / CIFAR-100 / SVHN

Nalisnick et al. Do Deep Generative Models Know What They Don't Know. ICLR 2019.
Ren et al. Likelihood Ratios for Out-of-Distribution Detection. NeurIPS 2019.

Novelty/Out-of-distribution Detection

Case 2: $f(\cdot)$ = a generative model from an unlabeled dataset $\mathcal{D} = \{\mathbf{x}_i\}$

1. How to learn a better representation $f(\cdot)$ more suitable for OOD detection?
2. How to define a detection score $s(\cdot)$ that maximally utilizes $f(\cdot)$?

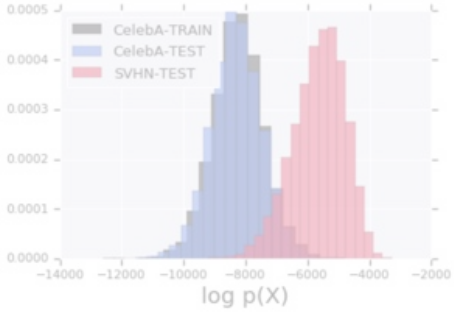
Ideally, a good likelihood model $p(\mathbf{X})$ should also represent a good $f(\cdot)$

- Unfortunately, ...
  1. ...
  2. Scaling up for a better likelihood model is usually much more challenging

At least for now, generative models also suffers from OODs

(a) Train on FashionMNIST, Test on MNIST

(b) Train on CIFAR-10, Test on SVHN



(c) Train on CelebA, Test on SVHN

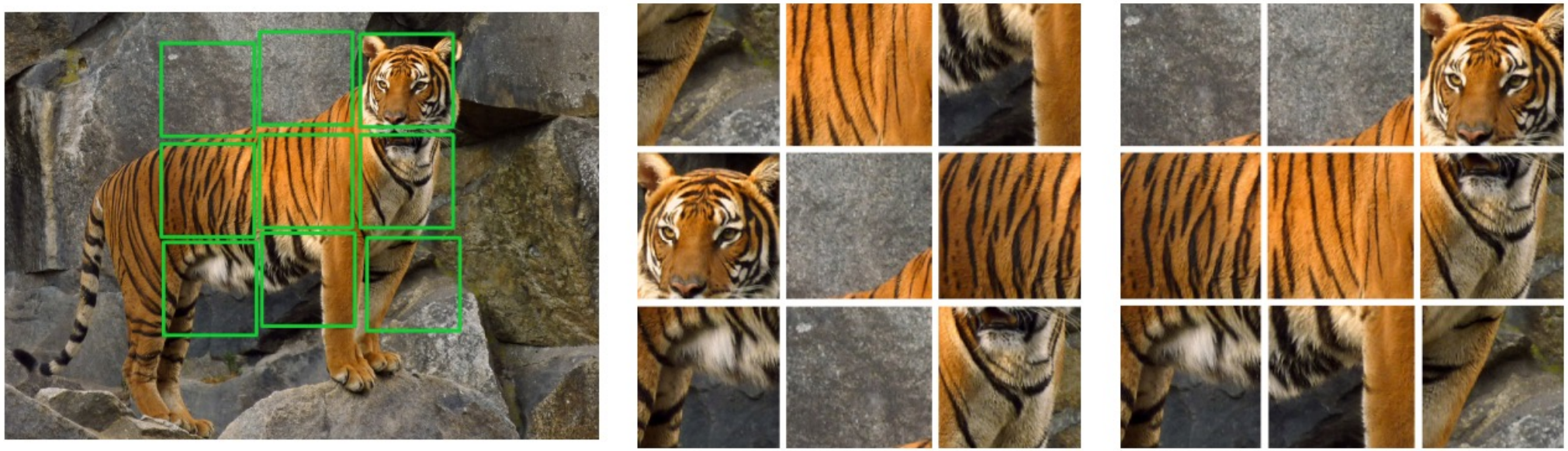
(d) Train on ImageNet,
Test on CIFAR-10 / CIFAR-100 / SVHN

Nalisnick et al. Do Deep Generative Models Know What They Don't Know. ICLR 2019.
Ren et al. Likelihood Ratios for Out-of-Distribution Detection. NeurIPS 2019.

Self-supervised Representation Learning (SSL)

- Learning unsupervised representation with self-supervision

- **Example 1:** Solving Jigsaw puzzles [Noroozi et al., 2016]

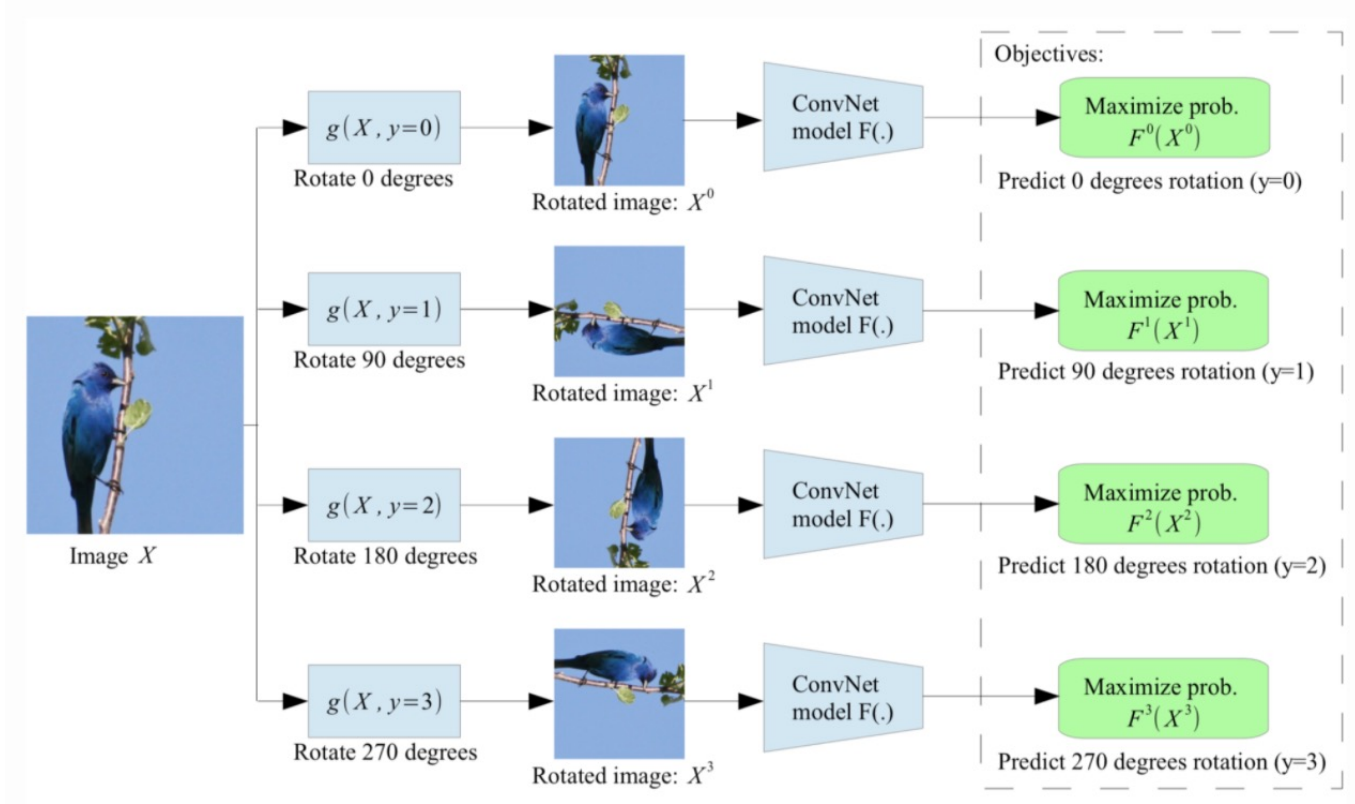


Noroozi et al. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. ECCV 2016.

Self-supervised Representation Learning (SSL)

- Learning unsupervised representation with self-supervision

- **Example 2:** Predicting rotation angles (RotNet) [Gidaris et al., 2016]



Gidaris et al. Unsupervised Representation Learning by Predicting Image Rotations. ICLR 2018.

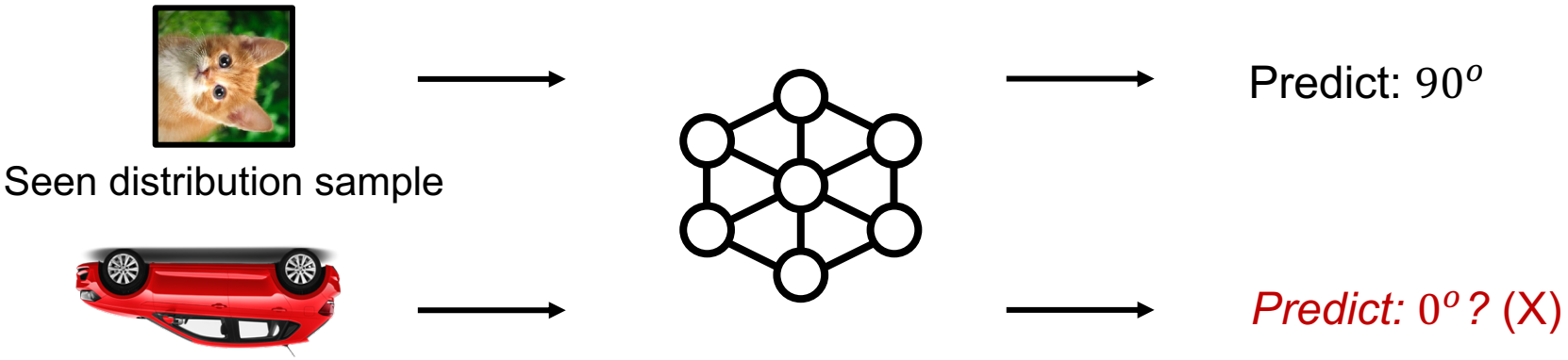
Self-supervised Representation Learning (SSL)

Hendrycks et al. (2019): **RotNet improves novelty detection**

$$\mathcal{L}_{\text{SS}}(x;\theta) = \frac{1}{4}\left[\sum_{r\in\{0°,90°,180°,270°\}} \mathcal{L}_{\text{CE}}(\texttt{one\_hot}(r), p_{\texttt{rot\_head}}(r \mid R_r(x)); \theta)\right]$$

- $f(\cdot)$ is trained to predict the rotation angle {0°, 90°, 180°, 270°} applied to the input
- $s(\cdot)$ is defined to detect samples those failed to predict the applied rotations

- **Intuition:** Predicting rotations are harder to be transferred to OODs



Hendrycks, Mazeika, Kadavath and Song. Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty. NeurIPS 2019.

Self-supervised Representation Learning (SSL)

Hendrycks et al. (2019): **RotNet improves novelty detection**

$$\mathcal{L}_{\text{SS}}(x;\theta) = \frac{1}{4}\left[\sum_{r\in\{0°,90°,180°,270°\}} \mathcal{L}_{\text{CE}}(\texttt{one\_hot}(r), p_{\texttt{rot\_head}}(r \mid R_r(x)); \theta)\right]$$

- The proposed $s(\cdot)$ for RotNet largely advances AUROC on challenging one-class CIFAR-10

							w/ an external OOD training data	
	OC-SVM	DeepSVDD	Geometric	RotNet	DIM	IIC	Supervised (OE)	Ours
CIFAR-10 classes: Airplane	65.6	61.7	76.2	71.9	72.6	68.4	87.6	77.5
Automobile	40.9	65.9	84.8	94.5	52.3	89.4	93.9	96.9
Bird	65.3	50.8	77.1	78.4	60.5	49.8	78.6	87.3
Cat	50.1	59.1	73.2	70.0	53.9	65.3	79.9	80.9
Deer	75.2	60.9	82.8	77.2	66.7	60.5	81.7	92.7
Dog	51.2	65.7	84.8	86.6	51.0	59.1	85.6	90.2
Frog	71.8	67.7	82.0	81.6	62.7	49.3	93.3	90.9
Horse	51.2	67.3	88.7	93.7	59.2	74.8	87.9	96.5
Ship	67.9	75.9	89.5	90.7	52.8	81.8	92.6	95.2
Truck	48.5	73.1	83.4	88.8	47.6	75.7	92.1	93.3
Mean	58.8	64.8	82.3	83.3	57.9	67.4	87.3	90.1

Hendrycks, Mazeika, Kadavath and Song. Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty. NeurIPS 2019.

Self-supervised Representation Learning (SSL)

Hendrycks et al. (2019): **RotNet improves novelty detection**

$$\mathcal{L}_{\text{SS}}(x;\theta) = \frac{1}{4}\left[\sum_{r\in\{0°,90°,180°,270°\}} \mathcal{L}_{\text{CE}}(\texttt{one\_hot}(r), p_{\texttt{rot\_head}}(r \mid R_r(x)); \theta)\right]$$

- Th

Would this intuition still hold to more advanced SSL frameworks?
⇒ We examine a novelty detection from contrastive learning

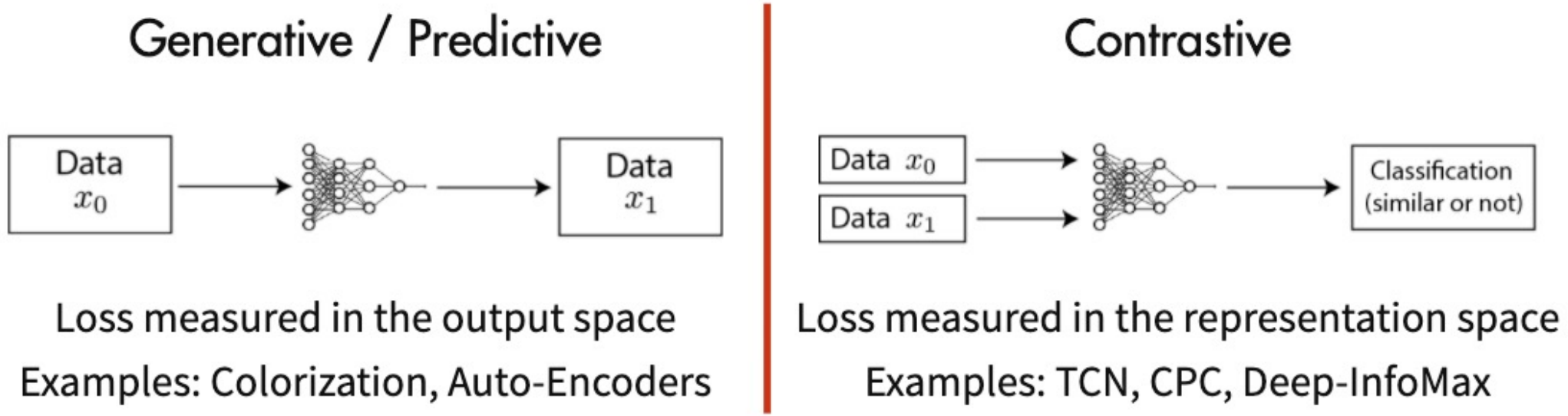
CIFAR-10 classes

Airplane	65.6	61.7	76.2	71.9	72.6	68.4	87.6	77.5
Automobile	40.9	65.9	84.8	94.5	52.3	89.4	93.9	96.9
Bird	65.3	50.8	77.1	78.4	60.5	49.8	78.6	87.3
Cat	50.1	59.1	73.2	70.0	53.9	65.3	79.9	80.9
Deer	75.2	60.9	82.8	77.2	66.7	60.5	81.7	92.7
Dog	51.2	65.7	84.8	86.6	51.0	59.1	85.6	90.2
Frog	71.8	67.7	82.0	81.6	62.7	49.3	93.3	90.9
Horse	51.2	67.3	88.7	93.7	59.2	74.8	87.9	96.5
Ship	67.9	75.9	89.5	90.7	52.8	81.8	92.6	95.2
Truck	48.5	73.1	83.4	88.8	47.6	75.7	92.1	93.3
Mean	58.8	64.8	82.3	83.3	57.9	67.4	87.3	90.1

Hendrycks, Mazeika, Kadavath and Song. Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty. NeurIPS 2019.

Contrastive Representation Learning

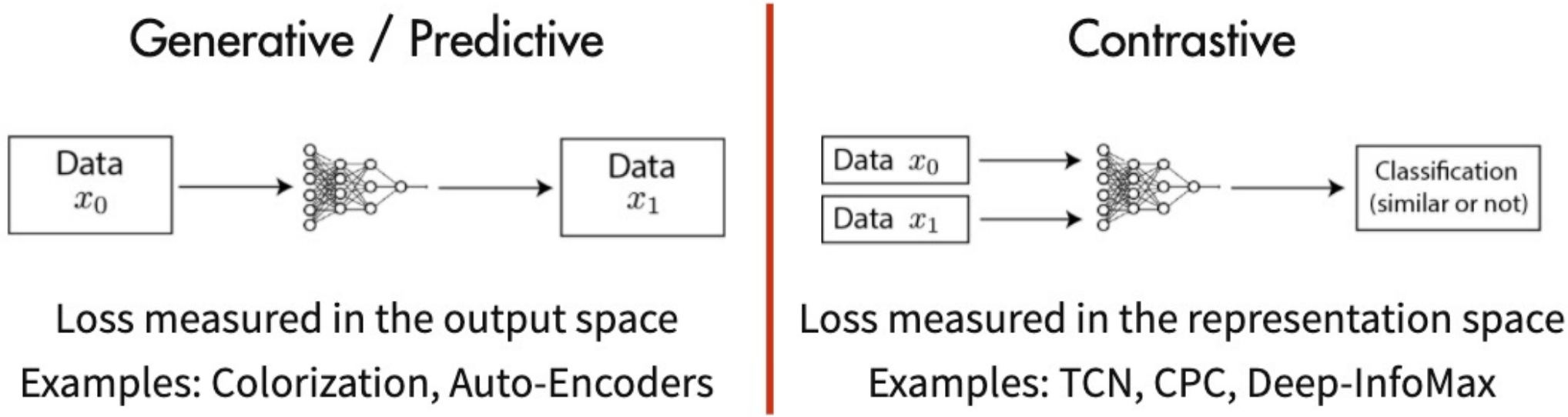
- Learning representation that **encodes the similarity between data points**



* source: https://ankeshanand.com/blog/2020/01/26/contrative-self-supervised-learning.html

Contrastive Representation Learning

- Learning representation that **encodes the similarity between data points**



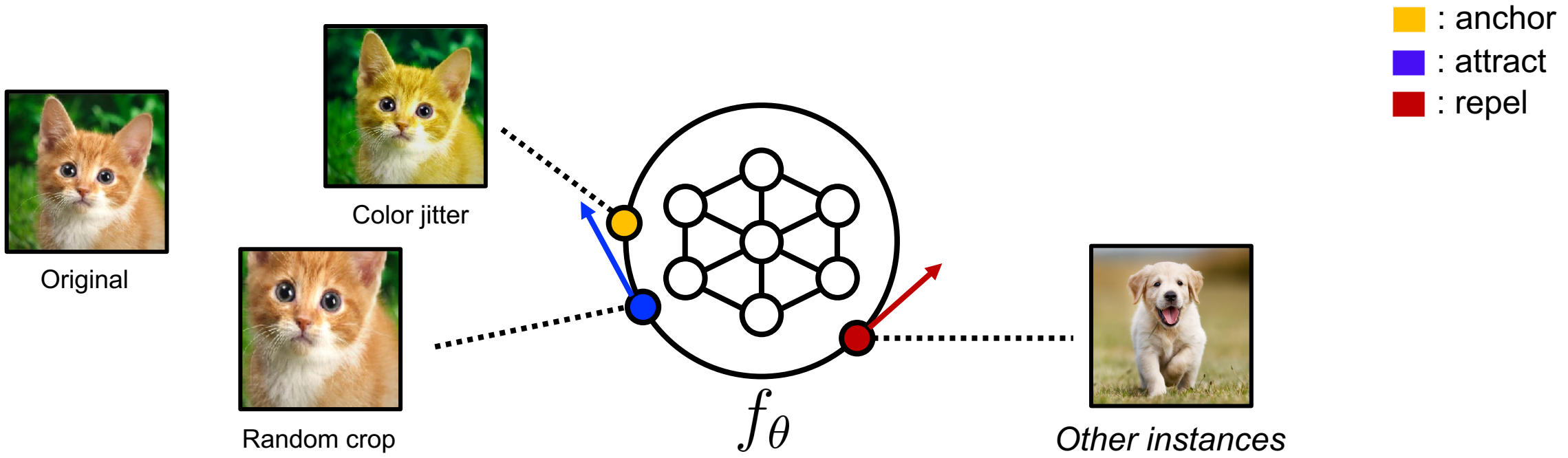
- A human prior defines positive & negative pairs w.r.t. a similarity score

$$\text{score}(f(x), f(x^+)) \gg \text{score}(f(x), f(x^-))$$

* source: https://ankeshanand.com/blog/2020/01/26/contrative-self-supervised-learning.html

Contrastive Representation Learning

- Learning representation that **encodes the similarity between data points**
- We focus on SimCLR, one of representatives of modern CL [Chen et al., 2020]:
  - **Attract (positive)** the same instances with *different data augmentations*
  - **Repel (negative)** all the other different instances

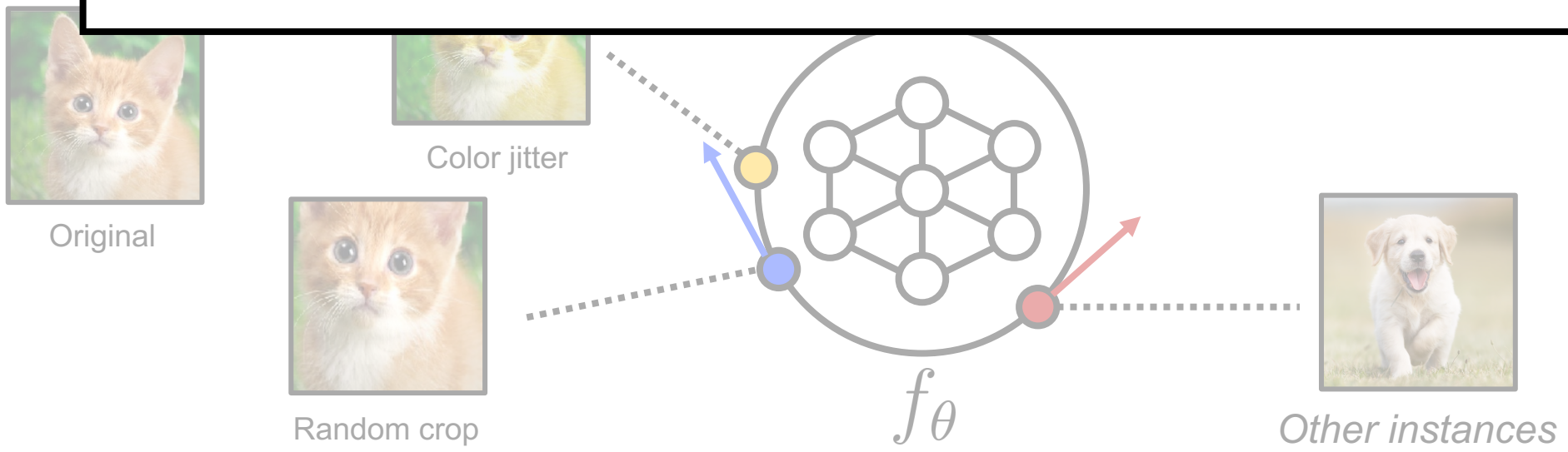


Chen et al. A simple framework for contrastive learning of visual representations. ICML 2020.

Contrastive Representation Learning

- Learning representation that **encodes the similarity between data points**
- We focus on SimCLR, one of representatives of modern CL [Chen et al., 2020]:
  - **Attract (positive)** the same instances with *different data augmentations*
  -

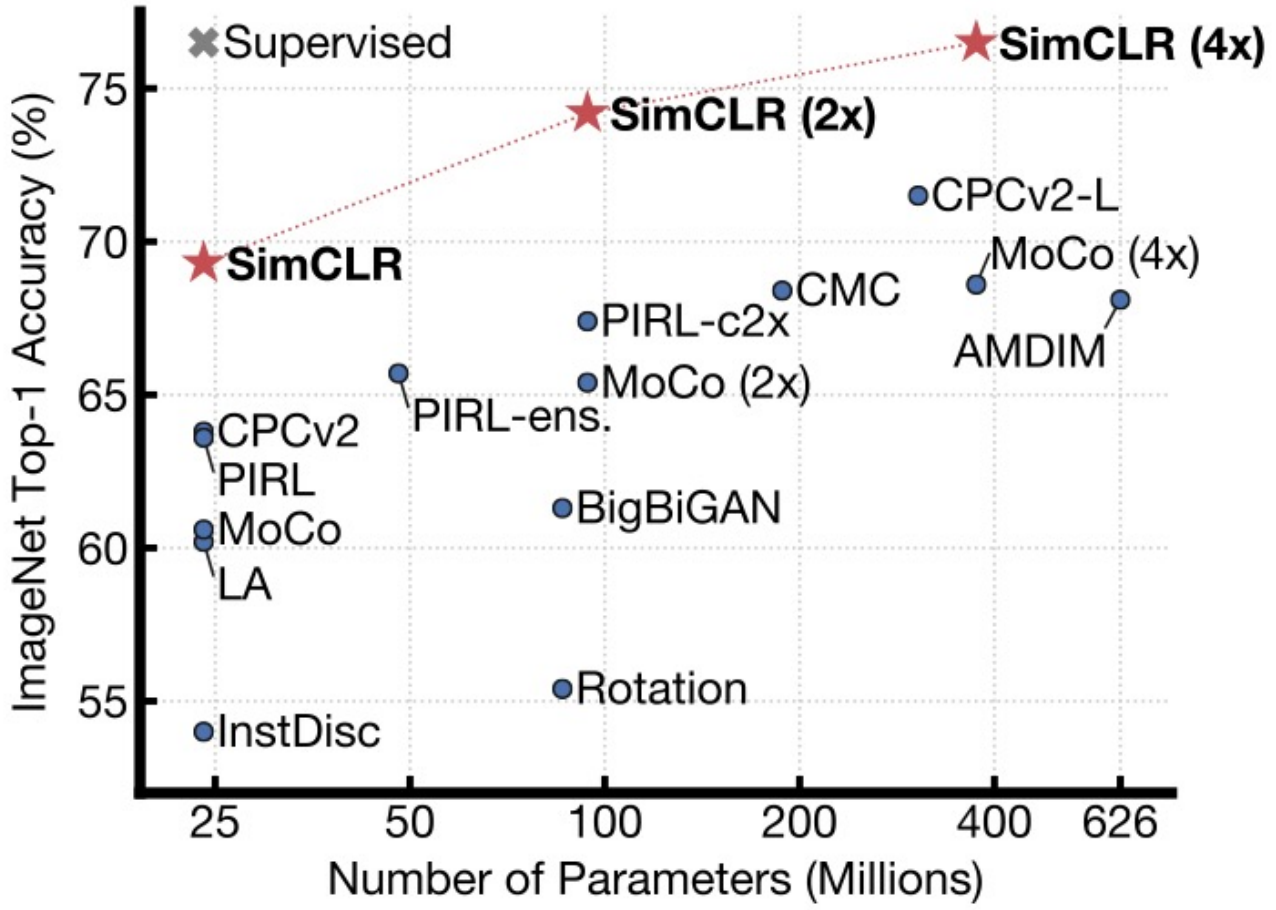
Learn representations by forcing those of augmented samples from the same instance "similar" and different instances "different".



Chen et al. A simple framework for contrastive learning of visual representations. ICML 2020.

Contrastive Representation Learning

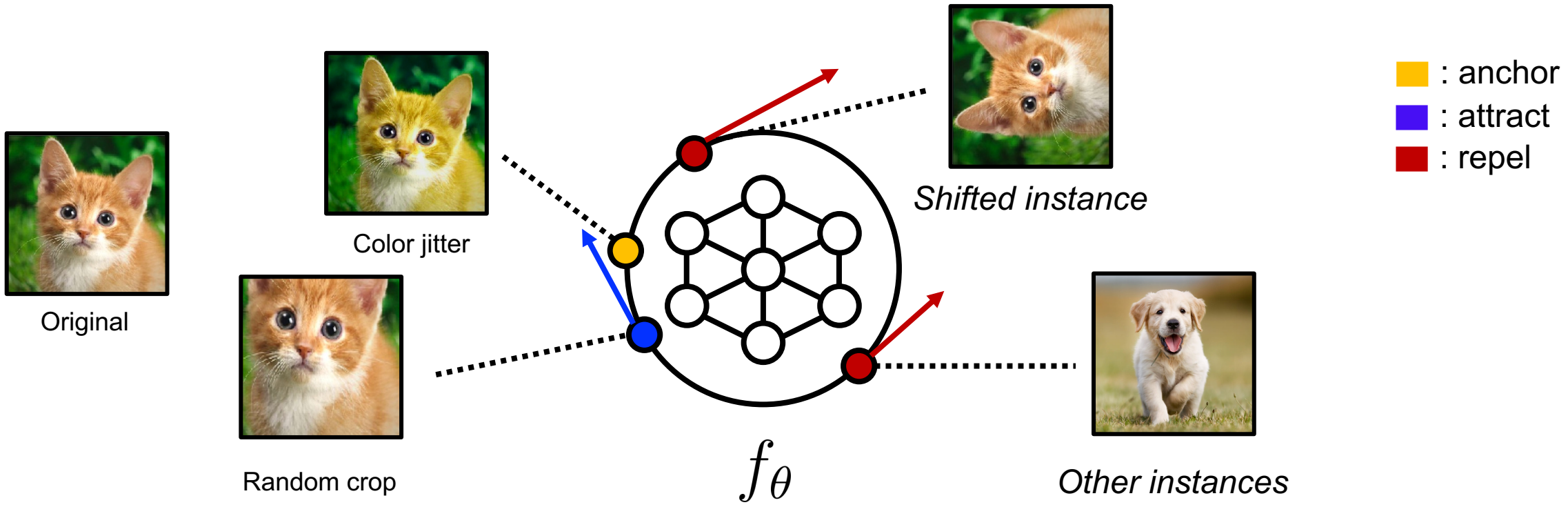
- SimCLR have largely closed the accuracy gap between un-/supervised learning



[1] Chen et al. A simple framework for contrastive learning of visual representations. ICML 2020.

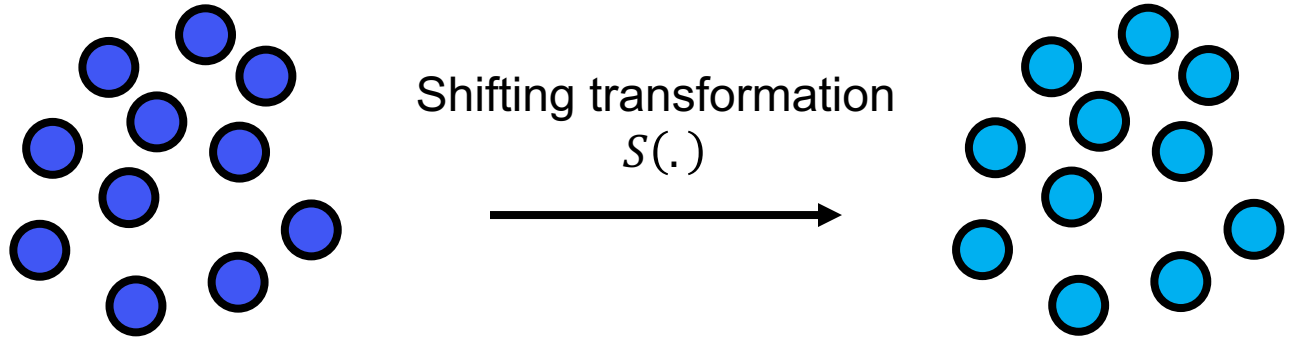
Summary: Contrasting Shifted Instances (CSI)

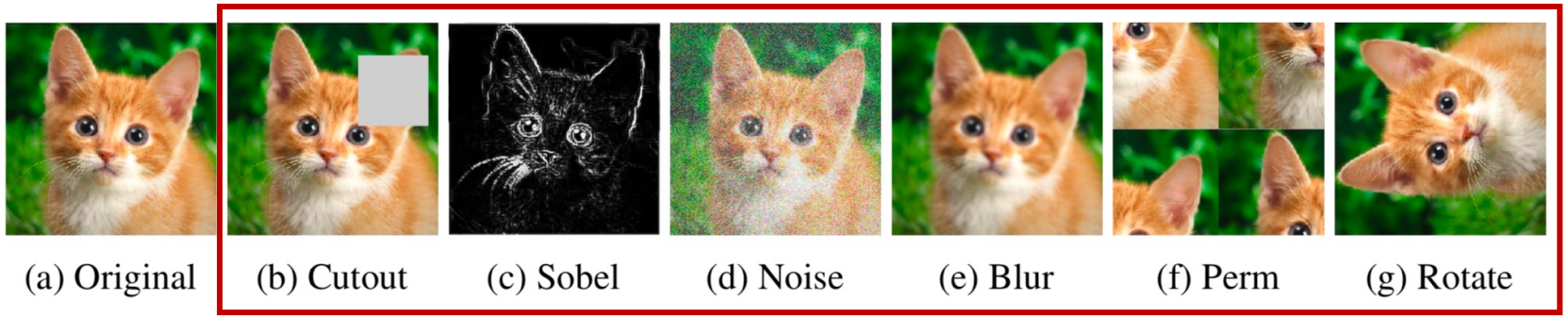
- We utilize the power of **contrastive learning** for OOD detection
- We further improve OOD detection by using **shifted instances**



Contrasting Shifted Instances (CSI): Representation

- We train the representation via **contrastive learning with shifted instances:**





Shifted instances = OOD-like augmented samples

(SimCLR uses color jitter, random crop, horizontal flip, grayscale)

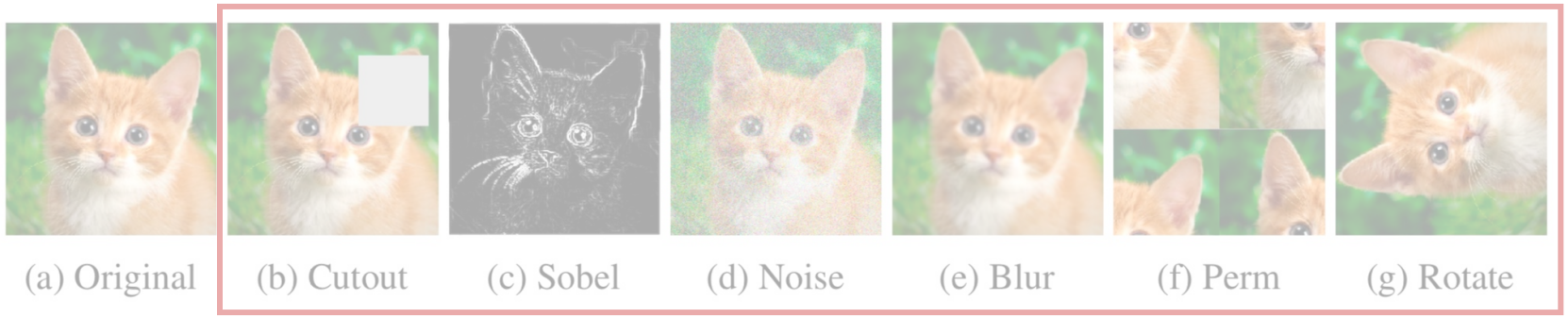
Contrasting Shifted Instances (CSI): Representation

- We train the representation via **contrastive learning with shifted instances:**

We force representations of "soft or IN-like" augmented samples (from the same instance) "similar"and "hard or OOD-like" ones "different".

Training samples

Shifted samples

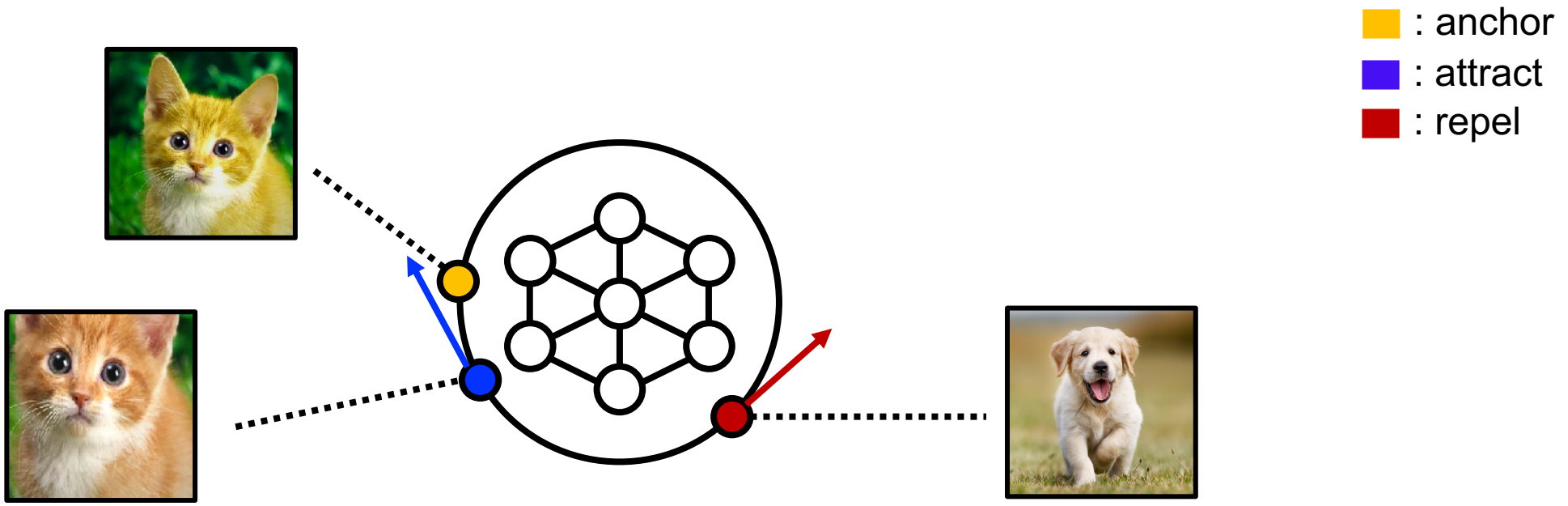


Shifted instances = OOD-like augmented samples

(SimCLR uses color jitter, random crop, horizontal flip, grayscale)

Contrasting Shifted Instances (CSI): Representation

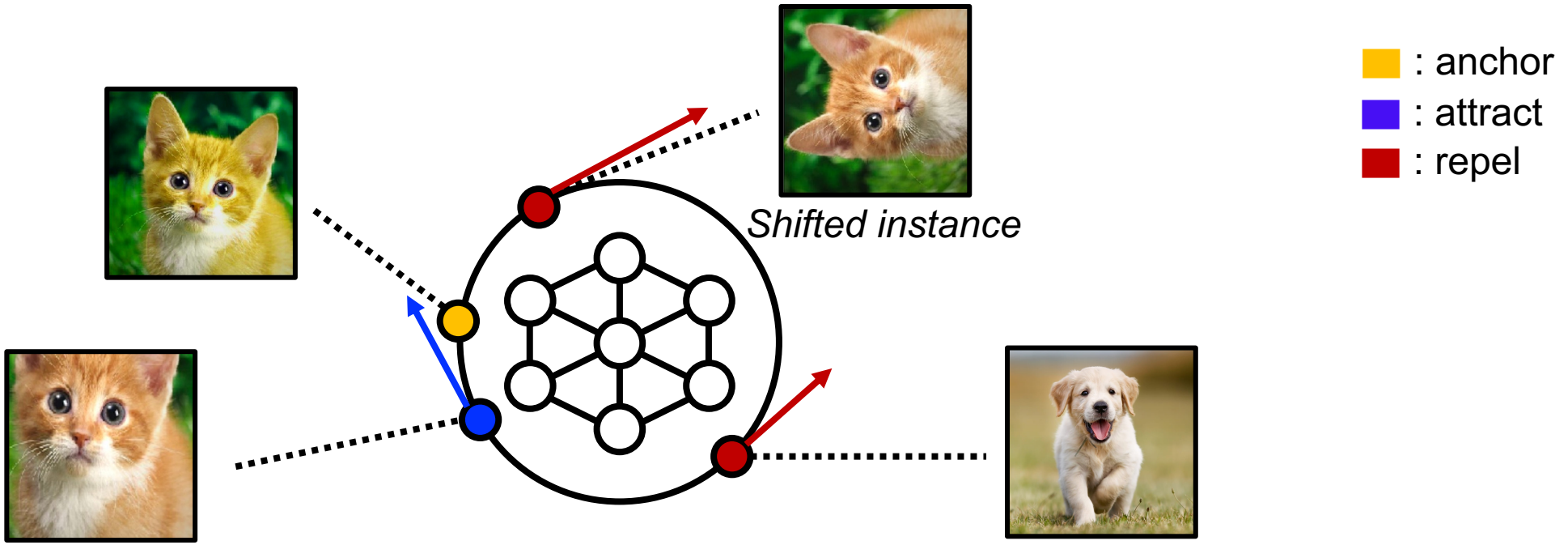
- We train the representation via **contrastive learning with shifted instances:**
  - We found **contrastive representation** [Chen et al., 2020] is already good at OOD detection



Chen et al. A simple framework for contrastive learning of visual representations. ICML 2020.

Contrasting Shifted Instances (CSI): Representation

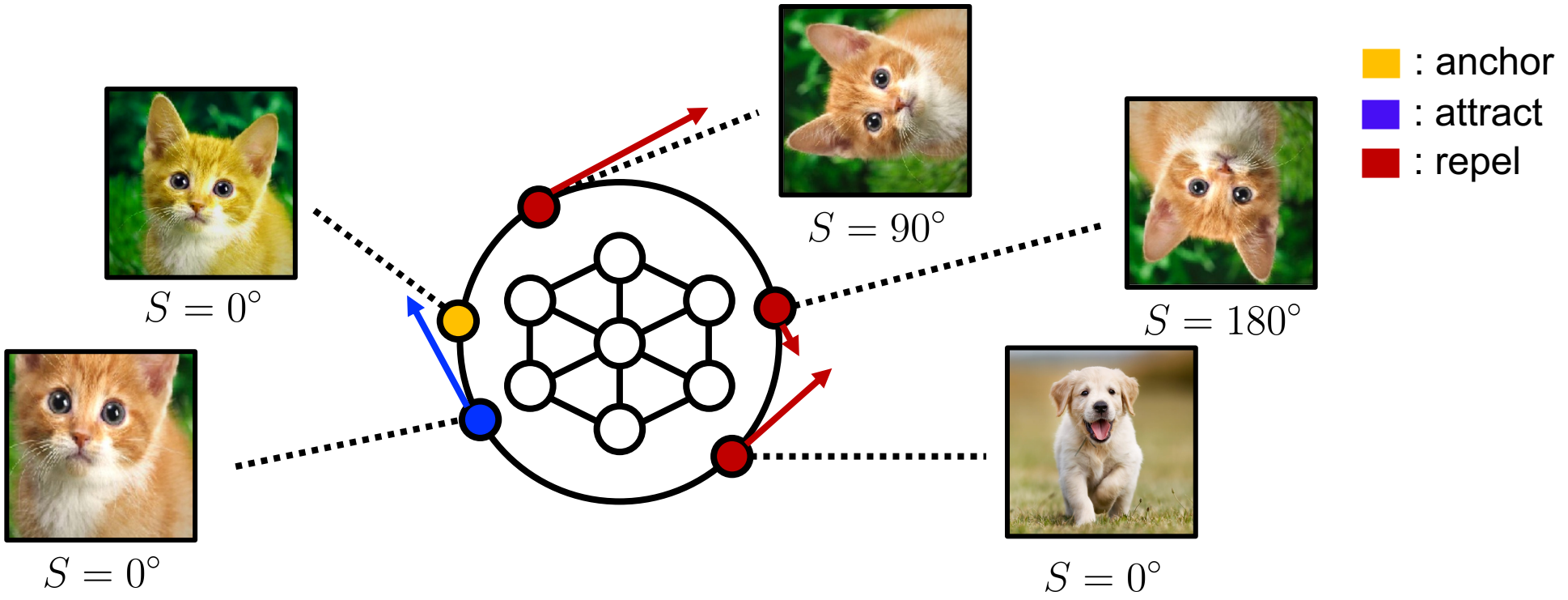
- We train the representation via **contrastive learning with shifted instances:**
  - We found **contrastive representation** [Chen et al., 2020] is already good at OOD detection
  - CSI further improves by **pushing the shifted samples** in addition to the different samples



Chen et al. A simple framework for contrastive learning of visual representations. ICML 2020.

Contrasting Shifted Instances (CSI): Representation

- We train the representation via **contrastive learning with shifted instances:**
  - We found **contrastive representation** [Chen et al., 2020] is already good at OOD detection
  - CSI further improves by **pushing the shifted samples** in addition to the different samples
  - Additionally **classify the shifting transformation**



Chen et al. A simple framework for contrastive learning of visual representations. ICML 2020.

Contrasting Shifted Instances (CSI): Detection Score

- Detection score for **contrastively learned representation:**

- Further improving the detection score by **utilizing the shifting transformation:**

Contrasting Shifted Instances (CSI): Detection Score

- Detection score for **contrastively learned representation:**
  - The ***cosine similarity*** to the nearest training sample
  - The ***norm*** of the representation

$$s_{\texttt{con}}(x; \{x_m\}) := \max_m \ \text{sim}(z(x_m), z(x)) \cdot \parallel z(x) \parallel .$$

score: cosine similarity * norm

- Further improving the detection score by **utilizing the shifting transformation:**

Contrasting Shifted Instances (CSI): Detection Score

- Detection score for **contrastively learned representation:**
  - The ***cosine similarity*** to the nearest training sample
  - The ***norm*** of the representation

$$s_{\text{con}}(x; \{x_m\}) := \max_m \ \text{sim}(z(x_m), z(x)) \cdot \| \ z(x) \ \| \ .$$

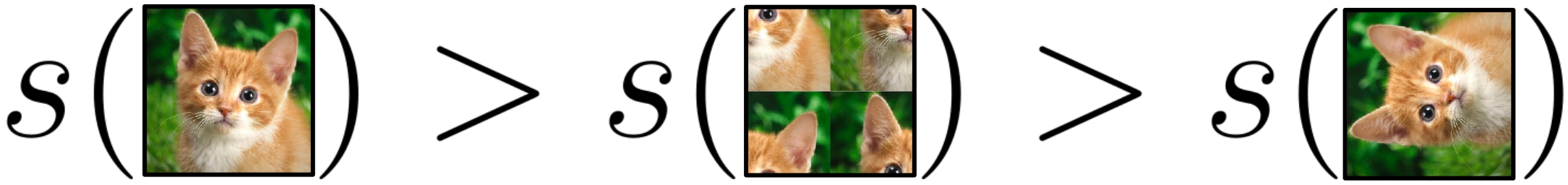
score: cosine similarity * norm

- Further improving the detection score by **utilizing the shifting transformation:**
  - $s_{\text{con-SI}}(x, \{x_m\})$: ensemble the score $s_{\text{con}}(x; \{x_m\})$ **over all shifting transformation**
  - $s_{\text{cls-SI}}(x)$: confidence of the **shifting transformation classifier**

$$s_{\text{CSI}}(x; \{x_m\}) := s_{\text{con-SI}}(x; \{x_m\}) + s_{\text{cls-SI}}(x)$$

Contrasting Shifted Instances (CSI): OOD-ness

- **OOD-ness**: How to choose the shifting transformation?
  - The transformation that generates the most **OOD-like yet semantically meaningful samples**
  - We choose the transformation with the high OOD-ness (AUROC on vanilla SimCLR)



Original sample

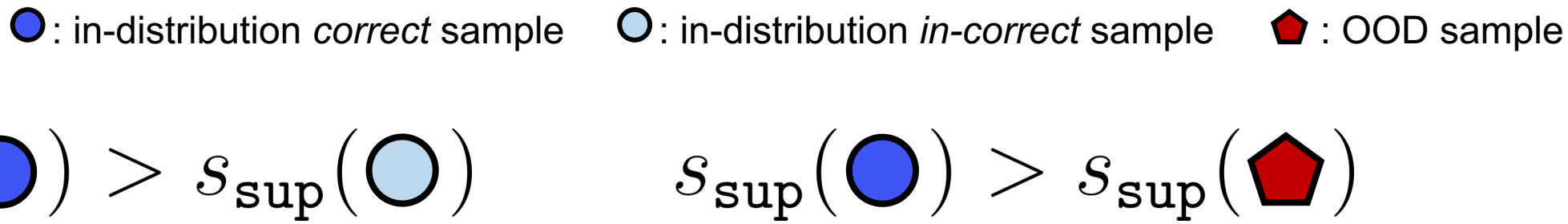
Low OOD-ness

High OOD-ness: Most OOD-like transformation

	Cutout	Sobel	Noise	Blur	Perm	Rotate
OOD-ness	79.5	69.2	74.4	76.0	83.8	85.2

Contrasting Shifted Instances (CSI): Extension

- We also **extend CSI** for training confidence-calibrated classifier [Lee et al., 2018]:
  - Accurate on predicting label y when input x is in-distribution
  - *Confidence* $s_{\text{sup}}(x) \coloneqq \max_y p(y|x)$ of the classifier is well-calibrated



Lee et al. Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples. ICLR 2018.
Khosla et al. Supervised contrastive learning. NeurIPS 2020.

Contrasting Shifted Instances (CSI): Extension

- We also **extend CSI** for training confidence-calibrated classifier [Lee et al., 2018]:
  - Accurate on predicting label y when input x is in-distribution
  - *Confidence* $s_{\text{sup}}(x) \coloneqq \max_y p(y|x)$ of the classifier is well-calibrated

● : in-distribution *correct* sample ○ : in-distribution *in-correct* sample ⬟ : OOD sample

$$s_{\text{sup}}(\bullet) > s_{\text{sup}}(\circ) \qquad s_{\text{sup}}(\bullet) > s_{\text{sup}}(\pentagon)$$

- We adapt the **idea of CSI** to the supervised contrastive learning (SupCLR):
  - SupCLR [Khosla et al., 2020] contrasts samples in *class-wise*, instead of in instance-wise
  - Similar to CSI, sup-CSI consider **shifted instance as a different class's sample**

Lee et al. Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples. ICLR 2018.
Khosla et al. Supervised contrastive learning. NeurIPS 2020.

Experiments: Unlabeled one-class OOD

- **CSI** achieves the state-of-the-art performance in **all tested scenarios**:
  - For unlabeled one-class OOD detection, outperforms prior methods under **all classes**

(a) One-class CIFAR-10

Method	Network	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Mean
OC-SVM* [64]	-	65.6	40.9	65.3	50.1	75.2	51.2	71.8	51.2	67.9	48.5	58.8
DeepSVDD* [60]	LeNet	61.7	65.9	50.8	59.1	60.9	65.7	67.7	67.3	75.9	73.1	64.8
AnoGAN* [63]	DCGAN	67.1	54.7	52.9	54.5	65.1	60.3	58.5	62.5	75.8	66.5	61.8
OCGAN* [55]	OCGAN	75.7	53.1	64.0	62.0	72.3	62.0	72.3	57.5	82.0	55.4	65.7
Geom* [17]	WRN-16-8	74.7	95.7	78.1	72.4	87.8	87.8	83.4	95.5	93.3	91.3	86.0
Rot* [27]	WRN-16-4	71.9	94.5	78.4	70.0	77.2	86.6	81.6	93.7	90.7	88.8	83.3
Rot+Trans* [27]	WRN-16-4	77.5	96.9	87.3	80.9	92.7	90.2	90.9	96.5	95.2	93.3	90.1
GOAD* [2]	WRN-10-4	77.2	96.7	83.3	77.7	87.8	87.8	90.0	96.1	93.8	92.0	88.2
Rot [27]	ResNet-18	78.3±0.2	94.3±0.3	86.2±0.4	80.8±0.6	89.4±0.5	89.0±0.4	88.9±0.4	95.1±0.2	92.3±0.3	89.7±0.3	88.4
Rot+Trans [27]	ResNet-18	80.4±0.3	96.4±0.2	85.9±0.3	81.1±0.5	91.3±0.3	89.6±0.3	89.9±0.3	95.9±0.1	95.0±0.1	92.6±0.2	89.8
GOAD [2]	ResNet-18	75.5±0.3	94.1±0.3	81.8±0.5	72.0±0.3	83.7±0.9	84.4±0.3	82.9±0.8	93.9±0.3	92.9±0.3	89.5±0.2	85.1
CSI (ours)	ResNet-18	**89.9**±0.1	**99.1**±0.0	**93.1**±0.2	**86.4**±0.2	**93.9**±0.1	**93.2**±0.2	**95.1**±0.1	**98.7**±0.0	**97.9**±0.0	**95.5**±0.1	**94.3**

(b) One-class CIFAR-100 (super-class)

Method	Network	AUROC
OC-SVM* [64]	-	63.1
Geom* [17]	WRN-16-8	78.7
Rot [27]	ResNet-18	77.7
Rot+Trans [27]	ResNet-18	79.8
GOAD [2]	ResNet-18	74.5
CSI (ours)	ResNet-18	**89.6**

(c) One-class ImageNet-30

Method	Network	AUROC
Rot* [27]	ResNet-18	65.3
Rot+Trans* [27]	ResNet-18	77.9
Rot+Attn* [27]	ResNet-18	81.6
Rot+Trans+Attn* [27]	ResNet-18	84.8
Rot+Trans+Attn+Resize* [27]	ResNet-18	85.7
CSI (ours)	ResNet-18	**91.6**

Experiments: Unlabeled multi-class OOD

- **CSI** achieves the state-of-the-art performance in **all tested scenarios**:
  - For unlabeled multi-class OOD detection, outperforms prior methods under **all OOD datasets**

(a) Unlabeled CIFAR-10

Method	Network	CIFAR10 → SVHN	LSUN	ImageNet	LSUN (FIX)	ImageNet (FIX)	CIFAR-100	Interp.
Likelihood*	PixelCNN++	8.3	-	64.2	-	-	52.6	52.6
Likelihood*	Glow	8.3	-	66.3	-	-	58.2	58.2
Likelihood*	EBM	63.0	-	-	-	-	-	70.0
Likelihood Ratio* [55]	PixelCNN++	91.2	-	-	-	-	-	-
Input Complexity* [61]	PixelCNN++	92.9	-	58.9	-	-	53.5	-
Input Complexity* [61]	Glow	95.0	-	71.6	-	-	73.6	-
Rot [25]	ResNet-18	97.6±0.2	89.2±0.7	90.5±0.3	77.7±0.3	83.2±0.1	79.0±0.1	64.0±0.3
Rot+Trans [25]	ResNet-18	97.8±0.2	92.8±0.9	94.2±0.7	81.6±0.4	86.7±0.1	82.3±0.2	68.1±0.8
GOAD [2]	ResNet-18	96.3±0.2	89.3±1.5	91.8±1.2	78.8±0.3	83.3±0.1	77.2±0.3	59.4±1.1
CSI (ours)	ResNet-18	**99.8**±0.0	**97.5**±0.3	**97.6**±0.3	**90.3**±0.3	**93.3**±0.1	**89.2**±0.1	**79.3**±0.2

(b) Unlabeled ImageNet-30

Method	Network	ImageNet-30 → CUB-200	Dogs	Pets	Flowers	Food-101	Places-365	Caltech-256	DTD
Rot [25]	ResNet-18	76.5±0.7	77.2±0.5	70.0±0.5	87.2±0.2	72.7±1.5	52.6±1.4	70.9±0.1	89.9±0.5
Rot+Trans [25]	ResNet-18	74.5±0.5	77.8±1.1	70.0±0.8	86.3±0.3	71.6±1.4	53.1±1.7	70.0±0.2	89.4±0.6
GOAD [2]	ResNet-18	71.5±1.4	74.3±1.6	65.5±1.3	82.8±1.4	68.7±0.7	51.0±1.1	67.4±0.8	87.5±0.8
CSI (ours)	ResNet-18	**90.5**±0.1	**97.1**±0.1	**85.2**±0.2	**94.7**±0.4	**89.2**±0.3	**78.3**±0.3	**87.1**±0.1	**96.9**±0.1

Experiments: Labeled multi-class OOD

- **CSI** achieves the state-of-the-art performance in **all tested scenarios**:
  - For labeled multi-class OOD detection, outperforms prior methods under **all OOD datasets**

(a) Labeled CIFAR-10

			CIFAR10 →						
Train method	Test acc.	ECE	SVHN	LSUN	ImageNet	LSUN (FIX)	ImageNet (FIX)	CIFAR100	Interp.
Cross Entropy	93.0±0.2	6.44±0.2	88.6±0.9	90.7±0.5	88.3±0.6	87.5±0.3	87.4±0.3	85.8±0.3	75.4±0.7
SupCLR [30]	93.8±0.1	5.56±0.1	97.3±0.1	92.8±0.5	91.4±1.2	91.6±1.5	90.5±0.5	88.6±0.2	75.7±0.1
CSI (ours)	94.8±0.1	4.40±0.1	96.5±0.2	96.3±0.5	96.2±0.4	92.1±0.5	92.4±0.0	90.5±0.1	78.5±0.2
CSI-ens (ours)	**96.1**±0.1	**3.50**±0.1	**97.9**±0.1	**97.7**±0.4	**97.6**±0.3	**93.5**±0.4	**94.0**±0.1	**92.2**±0.1	**80.1**±0.3

(b) Labeled ImageNet-30

			ImageNet-30 →							
Train method	Test acc.	ECE	CUB-200	Dogs	Pets	Flowers	Food-101	Places-365	Caltech-256	DTD
Cross Entropy	94.3	5.08	88.0	96.7	95.0	89.7	79.8	90.5	90.6	90.1
SupCLR [30]	96.9	3.12	86.3	95.6	94.2	92.2	81.2	89.7	90.2	92.1
CSI (ours)	97.0	2.61	93.4	97.7	96.9	96.0	87.0	92.5	91.9	93.7
CSI-ens (ours)	**97.8**	**2.19**	**94.6**	**98.3**	**97.4**	**96.2**	**88.9**	**94.0**	**93.2**	**97.4**

Experiments: Ablation study

- We verified the effectiveness of **shifting transformation selection scheme**
  - **Higher OOD-ness valued transformation** leads to higher detection performance

(a) Original (b) Cutout (c) Sobel (d) Noise (e) Blur (f) Perm (g) Rotate

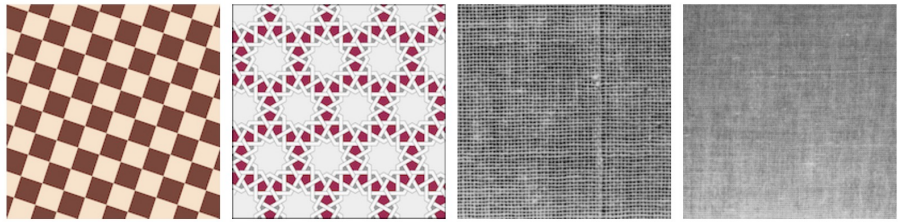
	Cutout	Sobel	Noise	Blur	Perm	Rotate
OOD-ness	79.5	69.2	74.4	76.0	83.8	85.2

Higher OOD-ness → Higher performance

Base		Cutout	Sobel	Noise	Blur	Perm	Rotate
87.9	+Align	84.3	85.0	85.5	88.0	73.1	76.5
	+Shift	88.5	88.3	89.3	89.2	90.7	94.3

Experiments: Ablation study

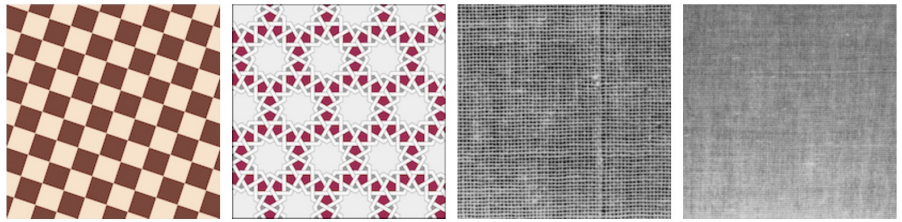
- We verified the effectiveness of **shifting transformation selection scheme**
  - **Higher OOD-ness valued transformation** leads to higher detection performance
  - Our method works on rotation-invariant datasets *i.e., rotation is not shifting transformation*



(a) OOD-ness		(b) AUROC		
Rot.	Noise	Base	CSI(R)	CSI(N)
50.6	**75.7**	70.3	65.9	**80.1**

Experiments: Ablation study

- We verified the effectiveness of **shifting transformation selection scheme**
  - **Higher OOD-ness valued transformation** leads to higher detection performance
  - Our method works on rotation-invariant datasets *i.e., rotation is not shifting transformation*



(a) OOD-ness		(b) AUROC		
Rot.	Noise	Base	CSI(R)	CSI(N)
50.6	**75.7**	70.3	65.9	**80.1**

- Each of the proposed components is complementary for AUROC

(a) Training objective

	SimCLR	Con.	Cls.	AUROC
$\mathcal{L}_{\text{SimCLR}}$ (2)	✓	-	-	87.9
$\mathcal{L}_{\text{con-SI}}$ (3)	✓	✓	-	91.6
$\mathcal{L}_{\text{cls-SI}}$ (4)	-	-	✓	88.6
$\mathcal{L}_{\text{CSI}}$ (5)	✓	✓	✓	**94.3**

(b) Detection score

	Con.	Cls.	Ensem.	AUROC
s_{con} (6)	✓	-	-	91.3
$s_{\text{con-SI}}$ (7)	✓	-	✓	93.3
$s_{\text{cls-SI}}$ (8)	-	✓	✓	93.8
s_{CSI} (9)	✓	✓	✓	**94.3**

Conclusion

- We propose **Contrasting Shifted Instances (CSI)** for OOD detection
  - We extend the power of contrastive learning for OOD detection
  - We further improve the OOD detection by utilizing shifting transformations

- CSI shows **outstanding performance** under various OOD detection scenarios

- We believe CSI would guide various future directions in **OOD detection** & **self-supervised learning** as an important baseline