Improving Transferability of Representations via Augmentation-Aware Self-Supervision

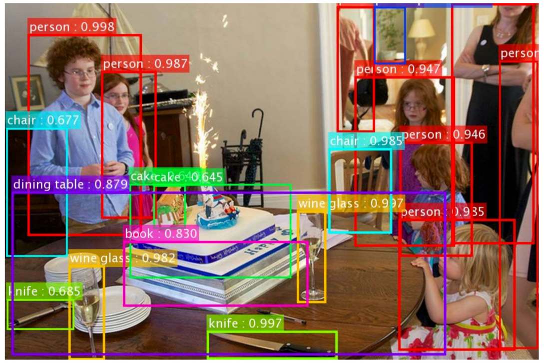
NeurIPS 2021

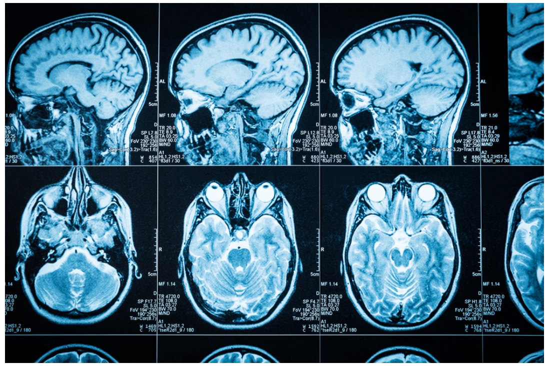
Hankook Lee[1] Kibok Lee[23] Kimin Lee[4] Honglak Lee[25] Jinwoo Shin[1]

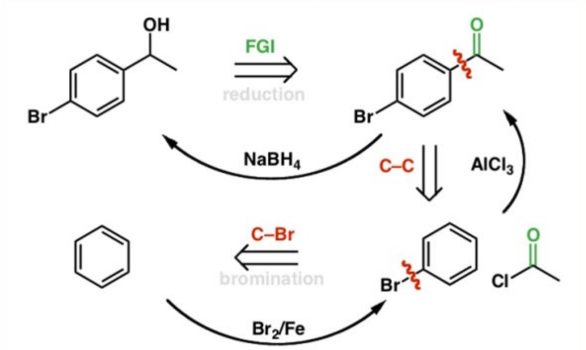
[1]Korea Advanced Institute of Science and Technology
[2]University of Michigan
[3]Amazon Web Services
[4]University of California, Berkeley
[5]LG AI Research

Unsupervised Representation Learning

- DNNs have achieved a **remarkable success** in various applications
  - They often **require a massive amount of manually labeled data**
  - The **annotation cost is often expensive** because
    - It is **time-consuming**: e.g., annotating bounding boxes
    - It requires **expert knowledge**: e.g., medical diagnosis and retrosynthesis





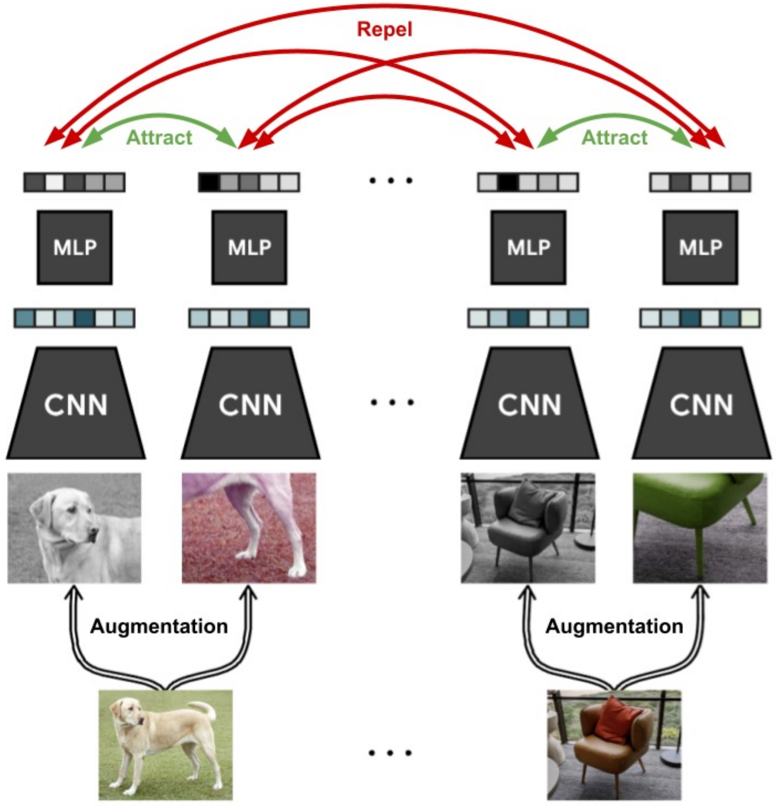


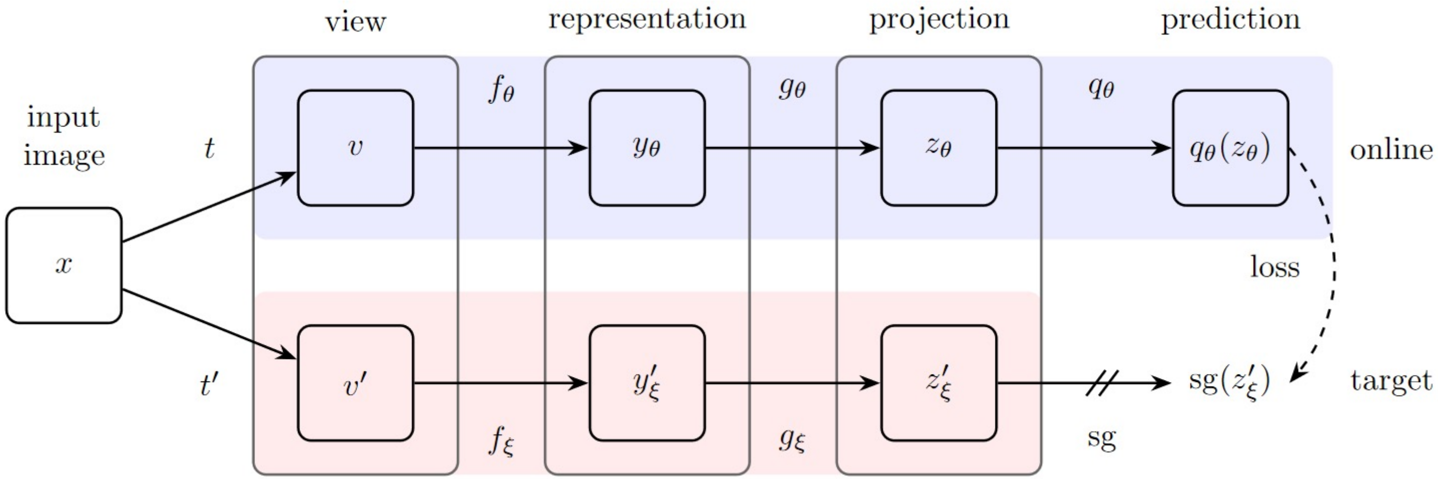
- Hence, **collecting unlabeled samples is easier** than doing labeled samples

- **Question:** How to utilize the unlabeled samples for representation learning?

Recent Advances in Self-supervised Learning

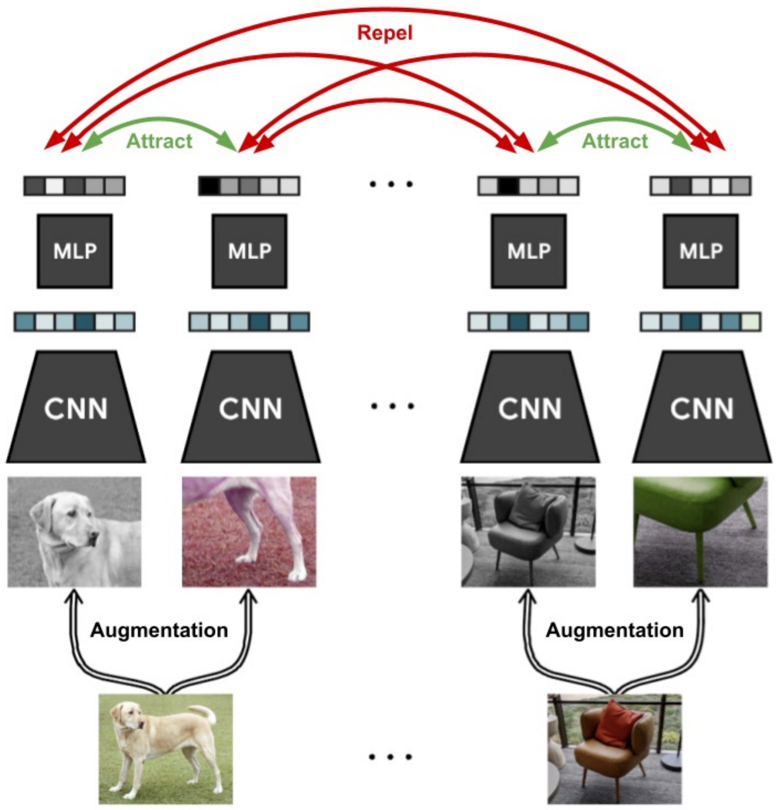
- State-of-the-art self-supervised learning methods have shown promising results
  - The SSL methods remarkably reduce the gap to supervised learning
  - They commonly **learn augmentation-invariant representations**





Recent Advances in Self-supervised Learning

- State-of-the-art self-supervised learning methods have shown promising results
  - The SSL methods remarkably reduce the gap to supervised learning
  - They commonly **learn augmentation-invariant representations**



Contrastive methods (e.g., SimCLR [1] and MoCo [2])

$$\mathcal{L} = -\log \frac{\exp(\mathrm{sim}(\mathbf{z}_1, \mathbf{z}_2)/\tau)}{\exp(\mathrm{sim}(\mathbf{z}_1, \mathbf{z}_2)/\tau) + \sum_{\mathbf{z}'} \exp(\mathrm{sim}(\mathbf{z}_1, \mathbf{z}')/\tau)}$$

Maximize $\mathrm{sim}(\mathbf{z}_1, \mathbf{z}_2)$ **Minimize** $\mathrm{sim}(\mathbf{z}_1, \mathbf{z}')$

$\mathbf{z}_1 = f(\mathbf{x}_1)$ $\mathbf{z}_2 = f(\mathbf{x}_2)$ $\mathbf{z}' = f(\mathbf{x}')$

$\mathbf{x}_1$

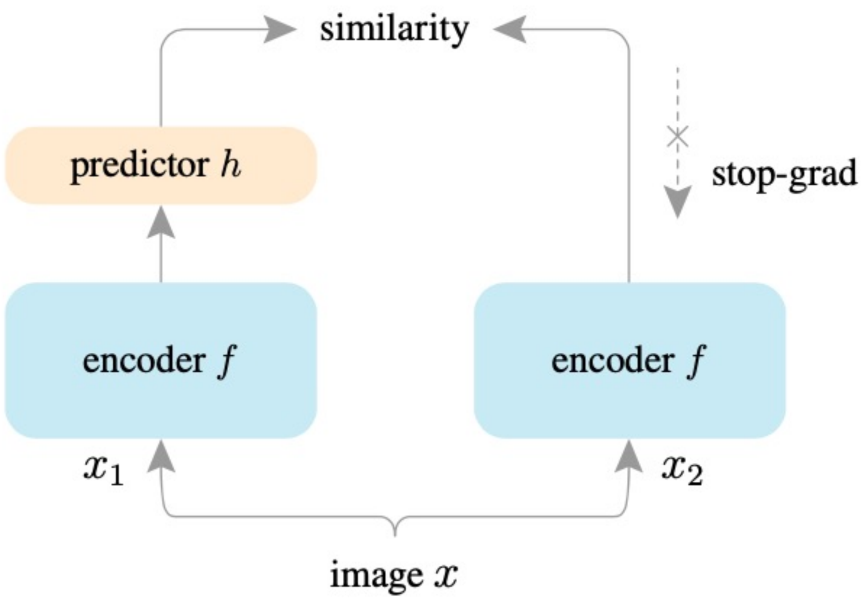
$\mathbf{x}_2$

$\mathbf{x}'$

[1] Chen et al., A Simple Framework for Contrastive Learning of Visual Representations, ICML 2020
[2] He et al., Momentum Contrast for Unsupervised Visual Representation Learning, CVPR 2020

Recent Advances in Self-supervised Learning

- State-of-the-art self-supervised learning methods have shown promising results
  - The SSL methods remarkably reduce the gap to supervised learning
  - They commonly **learn augmentation-invariant representations**



Non-contrastive methods (e.g., BYOL [3] and SimSiam [4])

$$\mathcal{L} = \|h(\mathbf{z}_1) - \texttt{stop\_grad}(\mathbf{z}_2)\|_2^2$$

$$\begin{aligned} \nabla\mathbb{E}[\mathcal{L}] &= \nabla\mathbb{E}[\|h^\star(\mathbf{z}_1) - \mathbf{z}_2\|_2^2] \\ &= \nabla\mathbb{E}[\|\mathbb{E}[\mathbf{z}_2|\mathbf{z}_1] - \mathbf{z}_2\|_2^2] \\ &= \nabla\mathbb{E}\left[\sum\nolimits_i \mathrm{Var}(\mathbf{z}_2^{(i)}|\mathbf{z}_1)\right] \end{aligned}$$

$\mathbf{z}_1 = f(\mathbf{x}_1)$ $\mathbf{z}_2 = f(\mathbf{x}_2)$

$\mathbf{x}_1$

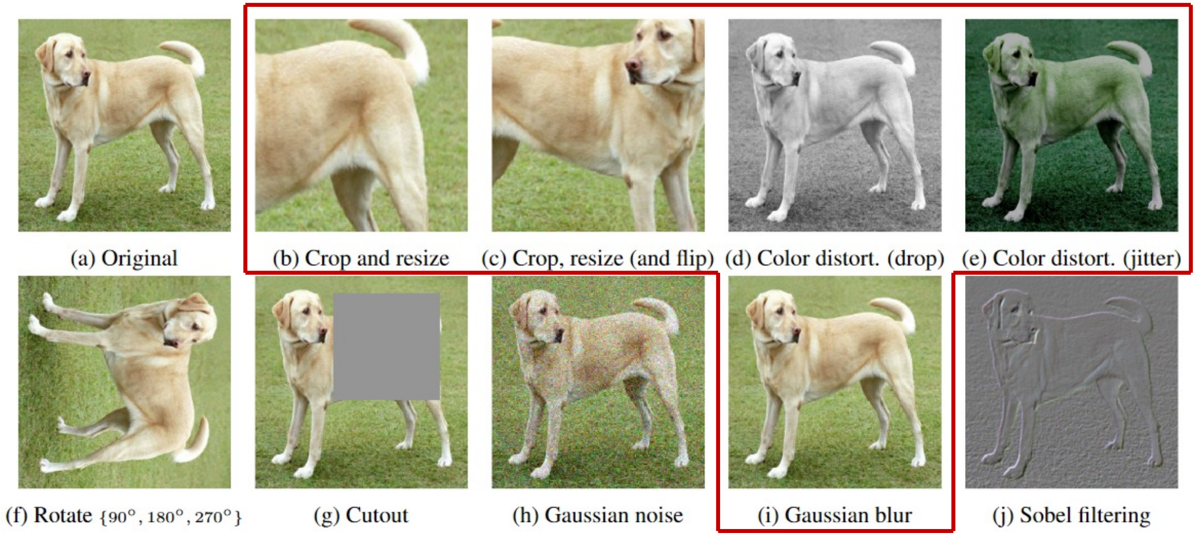
$\mathbf{x}_2$

[3] Grill et al., Bootstrap Your Own Latent: A New Approach to Self-supervised Learning, 2020
[4] Chen & He, Exploring Simple Siamese Representation Learning, 2020

Recent Advances in Self-supervised Learning

- State-of-the-art self-supervised learning methods have shown promising results
  - The SSL methods remarkably reduce the gap to supervised learning
  - They commonly **learn augmentation-invariant representations**
  - Augmentations:
    - **Geometric augmentations:** Cropping, Resizing, Flipping
    - **Color augmentations:** Color Jittering, Color Dropping, Gaussian Blurring

Commonly used augmentations for invariant representation learning



Motivation

- Total = (a) augmentation-invariant information + (b) augmentation-aware information

Original image

Augmentation

Color Dropping

Augmented image

Yellow flower

Flower of unknown color

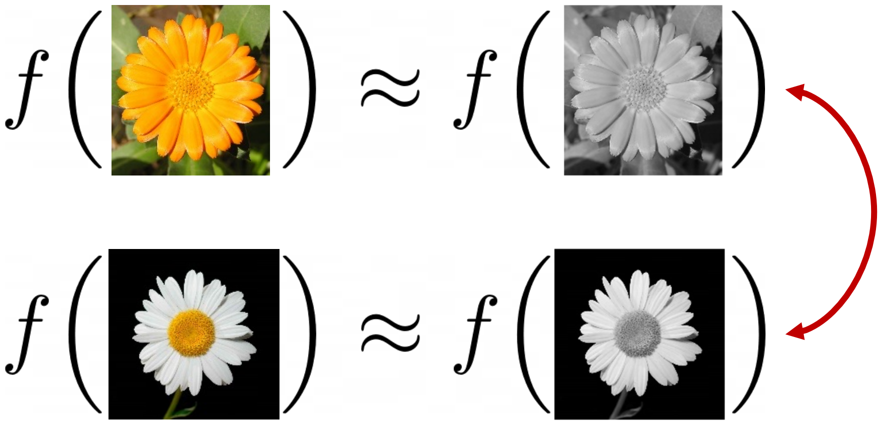
  - (a) augmentation-invariant information = **Flower**
  - (b) augmentation-aware information = **Yellow**

- **Q) Is augmentation-aware information not or less important?**

Motivation

- **Q) Is augmentation-aware information not or less important?**

- Learning augmentation-invariance may hurt performance in certain downstream tasks
  - Learning invariance to color augmentations (e.g., color dropping) forces the representations of color-modified and original images to be same as much as possible



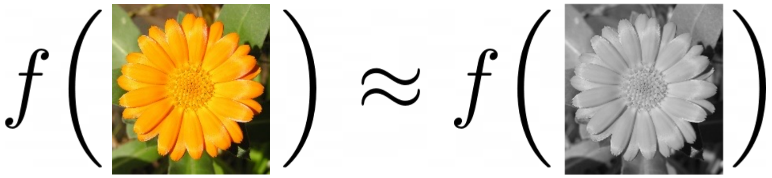
Which flower is yellow?

  - It degrades the representation qualities for color-sensitive downstream tasks such as flower classification

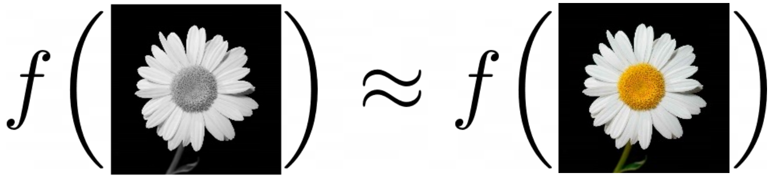
Motivation

- **Q) Is augmentation-aware information not or less important?**

- Learning augmentation-invariance may hurt performance in certain downstream tasks
  - Learning invariance to color augmentations (e.g., color dropping) forces the representations of color-modified and original images to be same as much as possible



v.s.

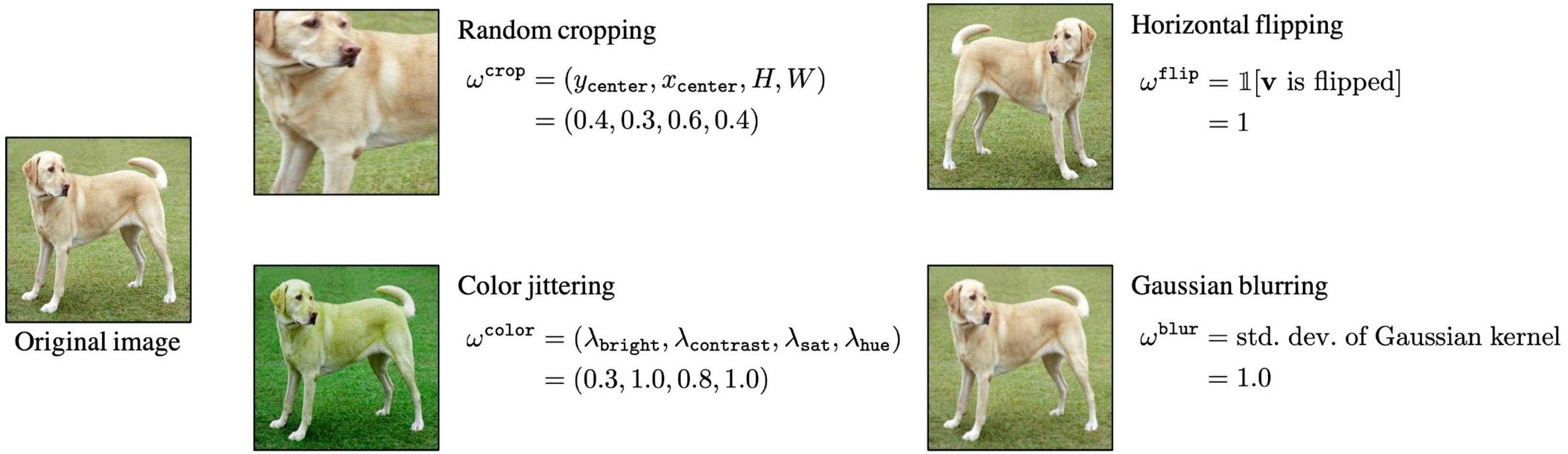


  - It degrades the representation qualities for color-sensitive downstream tasks such as flower classification

- **Q) How to learn more generalizable and transferable representations?**
- **Our goal** is to prevent information loss from learning augmentation-invariance, i.e., to **learn both augmentation-invariant and augmentation-aware representations**

AugSelf: Auxiliary Augmentation-aware Self-supervision

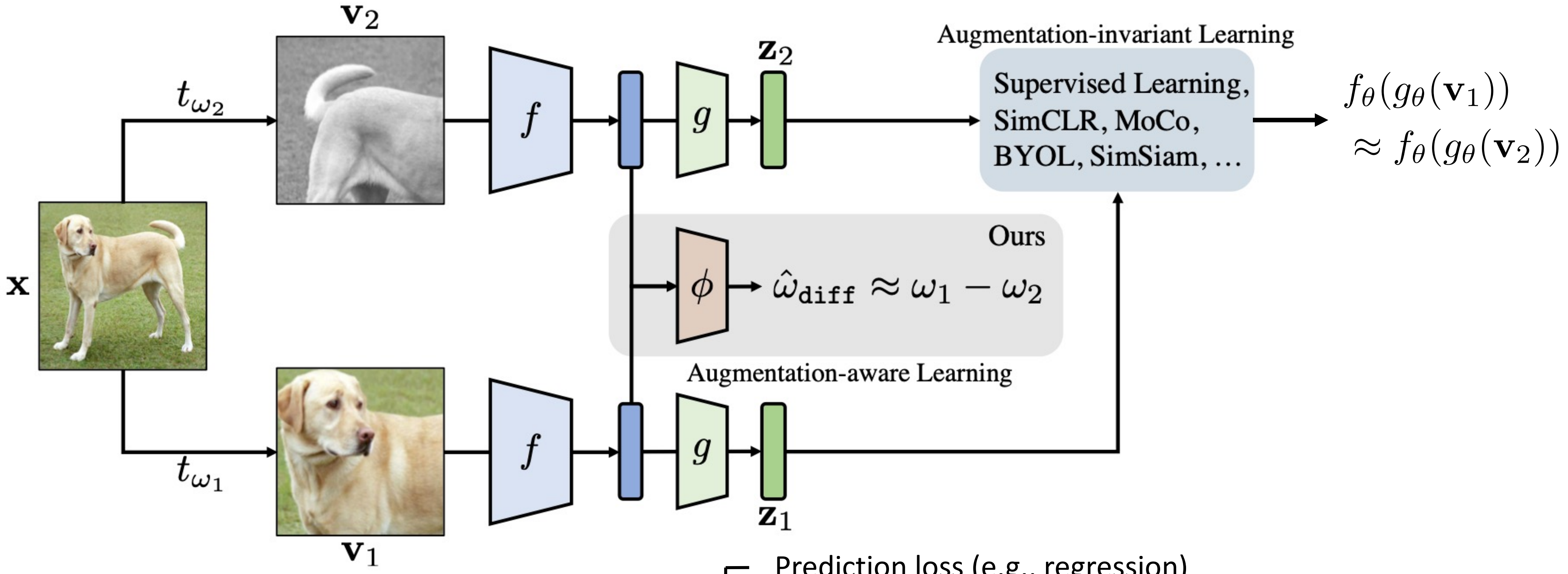
- **Notations**
  - Original image $\mathbf{x}$
  - Augmentation function t_ω where $\omega \sim \Omega$ is augmentation-specific parameter
  - Augmented view $\mathbf{v} = t_\omega(\mathbf{x})$
  - Examples:



  - Augmentation parameters ω explain how the image is modified

- Main idea is to **predict the augmentation parameters from augmented views**

AugSelf: Auxiliary Augmentation-aware Self-supervision



Prediction loss (e.g., regression)

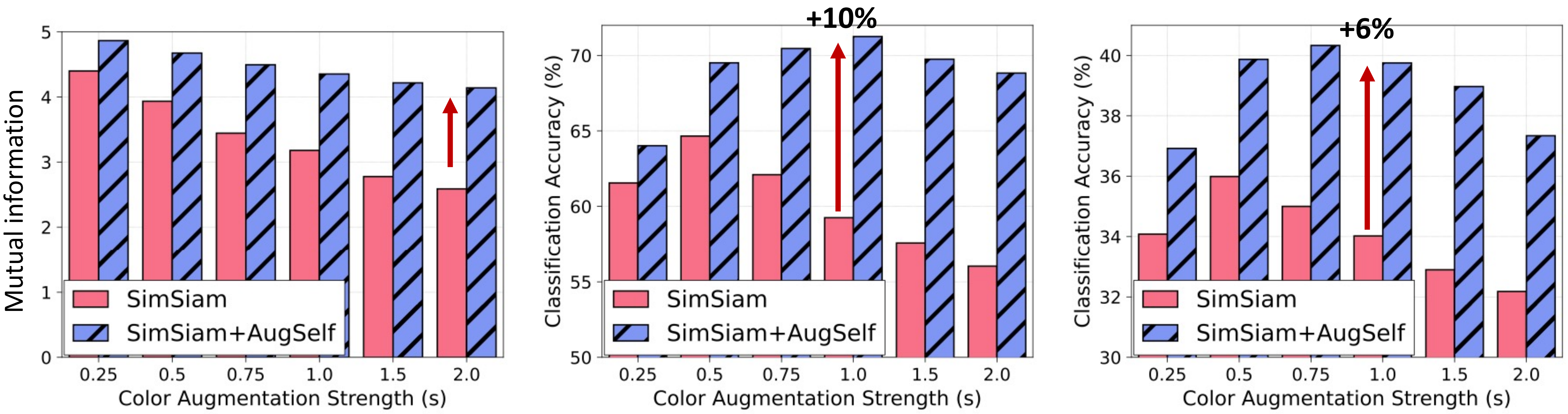
$$\mathcal{L}_{\texttt{AugSelf}}(\mathbf{x}, \omega_1, \omega_2; \theta) = \sum_{\texttt{aug} \in \mathcal{A}_{\texttt{AugSelf}}} \mathcal{L}_{\texttt{aug}}\left(\phi_\theta^{\texttt{aug}}(f_\theta(\mathbf{v}_1), f_\theta(\mathbf{v}_2)), \omega_{\texttt{diff}}^{\texttt{aug}}\right),$$

An augmentation type (e.g., crop, jittering)

- **AugSelf** learns to predict difference between augmentation parameters of two views
  - This prediction task encourages $f(x)$ to learn augmentation-aware information
  - This design allows to incorporate AugSelf into existing frameworks **without additional training costs**

Analysis: Mutual Information

- AugSelf preserves the augmentation-aware information
  - $I_{NCE}(C; Z)$ = the mutual information between color histogram (i.e., C) and representation (i.e., $Z = f(x)$)
  - AugSelf significantly improves the linear evaluation accuracy in the color-sensitive downstream tasks



(a) Mutual information (b) STL10→Flowers (c) STL10→Food

Ablation Study: All Information Is Useful

- Both color/geometric information is useful in various downstream tasks
  - Learn **color** information by predicting **Color Jittering** parameters
  - Learn **geometric** information by predicting **Random Cropping** parameters

Aug. parameters we predict

$\mathcal{A}_{\texttt{AugSelf}}$	STL10	CIFAR10	CIFAR100	Food	MIT67	Pets	Flowers
$\emptyset$	85.19	82.35	54.90	33.99	39.15	44.90	59.19
$\{\texttt{crop}\}$	**85.98**	82.82	55.78	35.68	43.21	47.10	62.05
$\{\texttt{color}\}$	85.55	**82.90**	58.11	40.32	43.56	47.85	71.08
$\{\texttt{crop}, \texttt{color}\}$	85.70	82.76	**58.65**	**41.58**	**45.67**	**48.42**	**72.18**

- The improvement depends on the characteristic of the downstream tasks
- Learning all information achieves best performance in most downstream tasks

Experimental Results: Fine-grained Classification Tasks

- AugSelf consistently improves Supervised Learning, SimSiam, MoCo in various settings
  - 11 fine-grained classification benchmarks

Method	CIFAR10	CIFAR100	Food	MIT67	Pets	Flowers	Caltech101	Cars	Aircraft	DTD	SUN397
ImageNet100-pretrained ResNet-50											
SimSiam	86.89	66.33	61.48	65.75	74.69	88.06	84.13	**48.20**	48.63	65.11	50.60
+ AugSelf	**88.80**	**70.27**	**65.63**	**67.76**	**76.34**	**90.70**	**85.30**	47.52	**49.76**	**67.29**	**52.28**
MoCo v2	84.60	61.60	59.37	61.64	70.08	82.43	77.25	33.86	**41.21**	64.47	46.50
+ AugSelf	**85.26**	**63.90**	**60.78**	**63.36**	**73.46**	**85.70**	**78.93**	**37.35**	39.47	**66.22**	**48.52**
Supervised	**86.16**	62.70	53.89	52.91	73.50	76.09	**77.53**	30.61	36.78	61.91	40.59
+ AugSelf	86.06	**63.77**	**55.84**	**54.63**	**74.81**	**78.22**	77.47	**31.26**	**38.02**	**62.07**	**41.49**
STL10-pretrained ResNet-18											
SimSiam	82.35	54.90	33.99	39.15	44.90	59.19	66.33	16.85	26.06	42.57	29.05
+ AugSelf	**82.76**	**58.65**	**41.58**	**45.67**	**48.42**	**72.18**	**72.75**	**21.17**	**33.17**	**47.02**	**34.14**
MoCo v2	81.18	53.75	33.69	39.01	42.34	61.01	64.15	16.09	26.63	41.20	28.50
+ AugSelf	**82.45**	**57.17**	**36.91**	**41.67**	**43.80**	**66.96**	**66.02**	**17.53**	**28.02**	**45.21**	**30.93**

Experimental Results: Fine-grained Classification Tasks

- AugSelf consistently improves Supervised Learning, SimSiam, MoCo in various settings
  - 11 fine-grained classification benchmarks

Method	CIFAR10	CIFAR100	Food	MIT67	Pets	Flowers	Caltech101	Cars	Aircraft	DTD	SUN397
ImageNet100-pretrained ResNet-50											
SimS											).60
+ Aug											**2.28**
MoCo											6.50
+ Aug											**8.52**
Super											).59
+ Aug											**1.49**
SimS											).05
+ Aug											**4.14**
MoCo v2											8.50
+ AugSelf	**82.45**	**57.17**	**36.91**	**41.67**	**43.80**	**66.96**	**66.02**	**17.53**	**28.02**	**45.21**	**30.93**

Method	AugSelf (ours)	STL10	CIFAR10	CIFAR100	Food	MIT67	Pets	Flowers
SimCLR [2]		84.87	78.93	48.94	31.97	36.82	43.18	56.20
	✓	**84.99**	**80.92**	**53.64**	**36.21**	**40.62**	**46.51**	**64.31**
BYOL [12]		86.73	82.66	55.94	37.30	42.78	50.21	66.89
	✓	**86.79**	**83.60**	**59.66**	**42.89**	**46.17**	**52.45**	**74.07**
SWAV [11]		82.21	81.60	52.00	29.78	36.69	37.68	53.01
	✓	**82.57**	**82.00**	**55.10**	**33.16**	**39.13**	**40.74**	**61.69**

Experimental Results: Few-shot Classification Tasks

- AugSelf consistently improves Supervised Learning, SimSiam, MoCo in various settings
  - 11 fine-grained classification benchmarks
  - 3 few-shot classification benchmarks

Method	FC100		CUB200		Plant Disease	
	(5, 1)	(5, 5)	(5, 1)	(5, 5)	(5, 1)	(5, 5) → 5-way 5-shot task
ImageNet100-pretrained ResNet-50						
SimSiam	36.19±0.36	50.36±0.38	45.56±0.47	62.48±0.48	75.72±0.46	89.94±0.31
+ AugSelf (ours)	**39.37±0.40**	**55.27±0.38**	**48.08±0.47**	**66.27±0.46**	**77.93±0.46**	**91.52±0.29**
MoCo v2	31.67±0.33	43.88±0.38	41.67±0.47	56.92±0.47	65.73±0.49	84.98±0.36
+ AugSelf (ours)	**35.02±0.36**	**48.77±0.39**	**44.17±0.48**	**57.35±0.48**	**71.80±0.47**	**87.81±0.33**
Supervised	33.15±0.33	46.59±0.37	46.57±0.48	63.69±0.46	68.95±0.47	88.77±0.30
+ AugSelf (ours)	**34.70±0.35**	**48.89±0.38**	**47.58±0.48**	**65.31±0.45**	**70.82±0.46**	**89.77±0.29**
STL10-pretrained ResNet-18						
SimSiam	36.72±0.35	51.49±0.36	37.97±0.43	50.61±0.45	58.13±0.50	75.98±0.40
+ AugSelf (ours)	**40.68±0.39**	**56.26±0.38**	**41.60±0.42**	**56.33±0.44**	**62.85±0.49**	**81.14±0.37**
MoCo v2	35.69±0.34	49.26±0.36	37.62±0.42	50.71±0.44	57.87±0.48	75.98±0.40
+ AugSelf (ours)	**39.66±0.39**	**55.58±0.39**	**38.33±0.41**	**51.93±0.44**	**60.78±0.50**	**78.76±0.38**

Experimental Results: Object Localization

- AugSelf consistently improves Supervised Learning, SimSiam, MoCo in various settings
  - 11 fine-grained classification benchmarks
  - 3 few-shot classification benchmarks
  - Object localization on CUB200 benchmark

Method	Error
SimSiam	0.00462
+ AugSelf	**0.00335**
MoCo	0.00487
+ AugSelf	**0.00429**
Supervised	0.00520
+ AugSelf	**0.00473**

Table 4: ℓ_2 errors of bounding box predictions on CUB200.

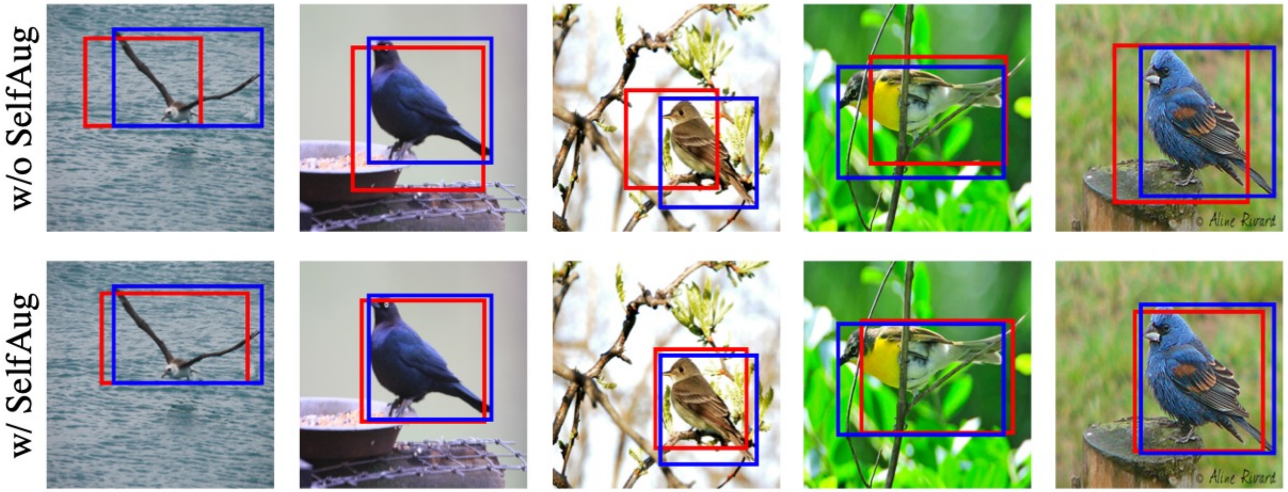
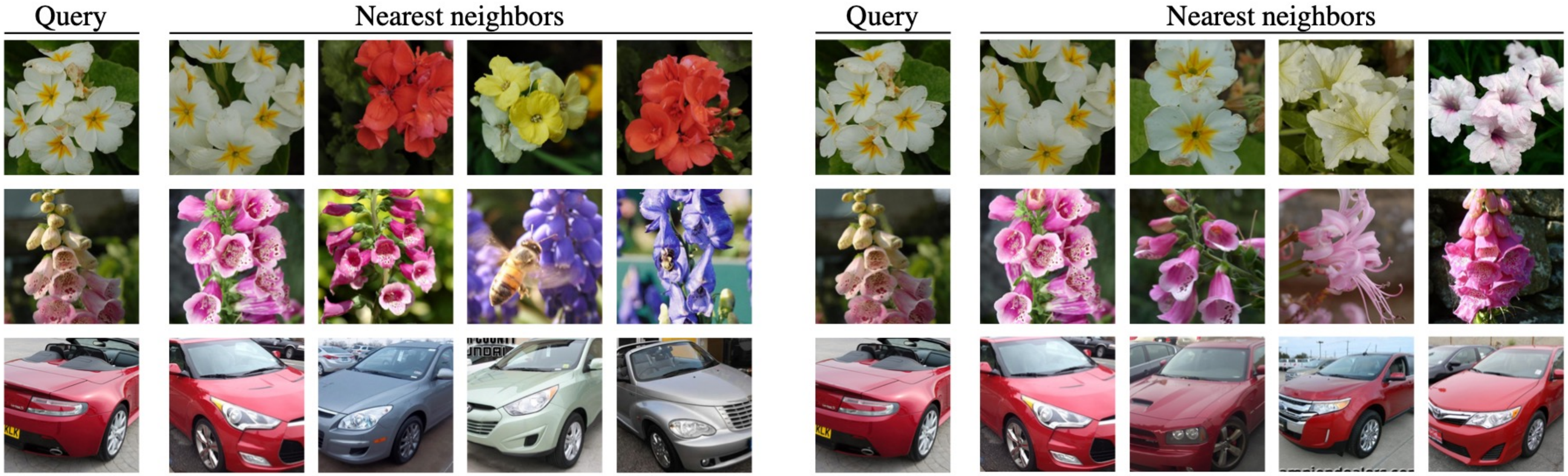


Figure 4: Examples of bounding box predictions on CUB200. Blue and red boxes are ground-truth and model prediction, respectively.

Experimental Results: Retrieval

- AugSelf consistently improves Supervised Learning, SimSiam, MoCo in various settings
  - 11 fine-grained classification benchmarks
  - 3 few-shot classification benchmarks
  - Object localization on CUB200 benchmark

- Quantitative analysis (based on retrieval)



(a) SimSiam (b) SimSiam + AugSelf

Conclusion

- We propose AugSelf for learning more transferable and generalizable representations
  - AugSelf encourages to preserve augmentation-aware information by learning the difference of augmentation parameters between two randomly augmented samples
  - AugSelf can easily be incorporated into recent state-of-the-art self-supervised learning methods with a negligible additional training cost
  - Extensive experiments demonstrate that AugSelf consistently improves the transferability of representations learned by supervised and unsupervised methods in various transfer learning scenarios

Thank you for your attention!