

Improving Transferability of Representations via Augmentation-Aware Self-Supervision

Hankook Lee¹, Kibok Lee^{2,3}, Kimin Lee⁴, Honglak Lee^{2,5}, Jinwoo Shin¹ ¹KAIST, ²University of Michigan, ³Amazon Web Services, ⁴University of California, Berkeley, ⁵LG AI Research

TL; DR. Learning augmentation-aware information by predicting the difference between two augmented samples improves the transferability of representations for various downstream tasks.

Background & Motivation

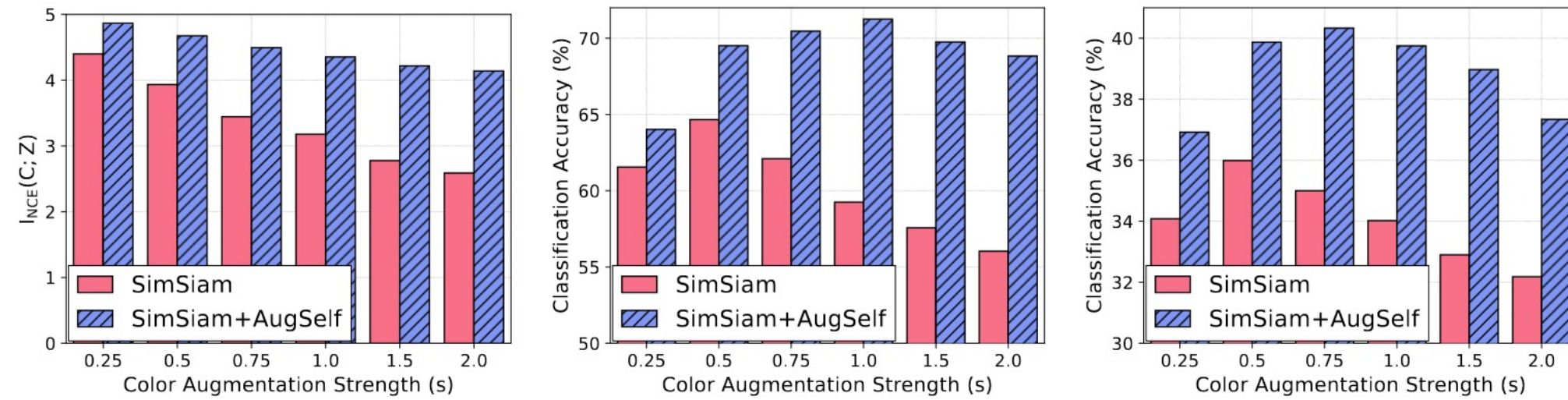
Self-supervised learning (SSL) learns representations via a pretext task that requires to predict self-supervision constructed from only input signals.

Recent SSL methods often aim at **learning invariance to data augmentations**

- Contrastive methods (e.g., MoCo [1], SimCLR [2])
- Negative-free methods (e.g., BYOL [3], SimSiam [4])
- Clustering-based methods (e.g., SwAV [5])

Question: is learning invariance to a given set of augmentations always beneficial to representation learning? To answer this question,

- Pretrain ResNet-18 on STL10 with varying color jittering strength s .
- Compute mutual information between the learned representation $Z=f(x)$ and color information $C(x)$ encoded by color histograms.
- Transfer the learned representation to color-sensitive downstream tasks.



(a) Mutual information

(b) STL10→Flowers

(c) STL10→Food

Observations:

- Stronger color augmentations \Rightarrow color-relevant information loss
- Less color information \Rightarrow performance drop in color-sensitive tasks

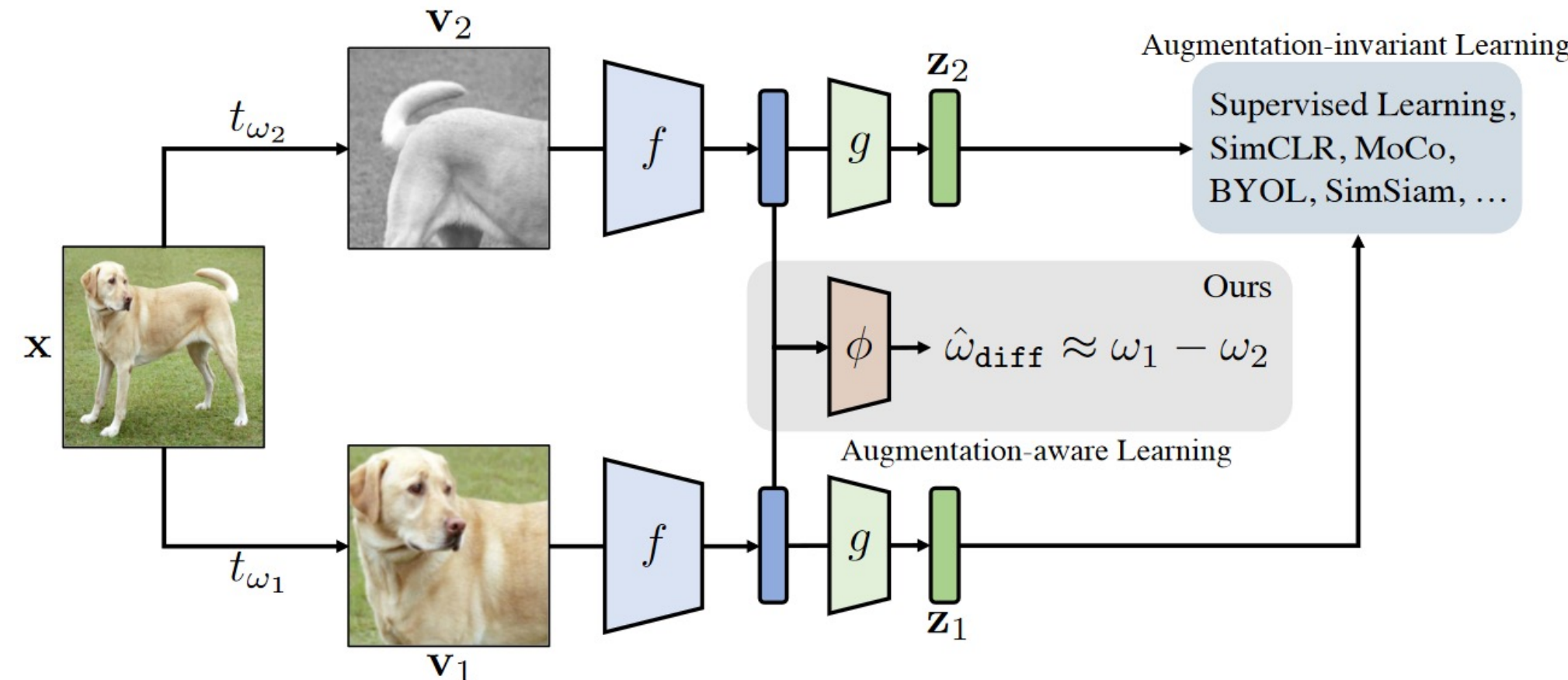
Research Question: how to prevent the information loss comes from learning the invariance?

Summary of Contribution

- For learning augmentation-aware information, we suggest to optimize an auxiliary self-supervised loss (**AugSelf**) that learns to predict difference between augmentation parameters of two randomly augmented samples.
- Extensive experiments demonstrate that (1) **AugSelf** can improve learned representations' transferability for various downstream tasks, and also (2) **AugSelf** can be easily incorporated with recent SSL methods with a negligible additional training cost.

Method

Notation. x is an original input image. t_ω is an augmentation function parameterized by ω . $v = t_\omega(x)$ is the augmented sample of x by t_ω . f is a CNN feature extractor such as ResNet. g is a projection MLP that is widely used in recent SSL methods [1-5]. ϕ is a prediction MLP for AugSelf.



For learning augmentation-aware information, we learn to predict the difference between two augmented samples. Formally, auxiliary augmentation-aware self-supervised loss (**AugSelf**) is defined by

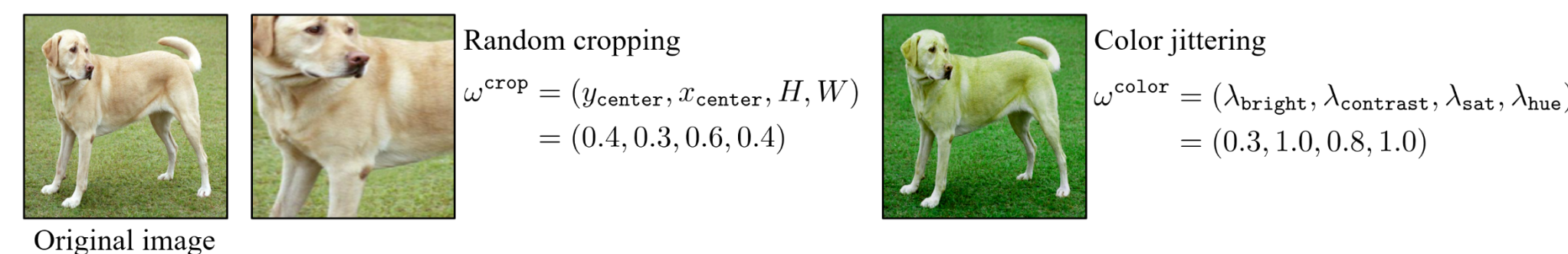
$$\mathcal{L}_{\text{AugSelf}}(x, \omega_1, \omega_2; \theta) = \sum_{\text{aug} \in \mathcal{A}_{\text{AugSelf}}} \mathcal{L}_{\text{aug}}(\phi_\theta^{\text{aug}}(f_\theta(v_1), f_\theta(v_2)), \omega_{\text{diff}}^{\text{aug}})$$

where $\omega_{\text{diff}}^{\text{aug}}$ is the difference between augmentation-specific parameters.

Benefits of AugSelf: it can ...

- preserve augmentation-aware information for downstream tasks
- be easily incorporated with [1-5] thanks to its self-supervision design

In this work, we mainly use $\mathcal{A}_{\text{AugSelf}} = \{\text{crop}, \text{color_jitter}\}$ and MSE for \mathcal{L}_{aug} .



References

- [1] He et al., Momentum Contrast for Unsupervised Visual Representation Learning, CVPR 2020
- [2] Chen et al., A Simple Framework for Contrastive Learning of Visual Representations, ICML 2020
- [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
- [5] Caron et al., Unsupervised Learning of Visual Features by Contrasting Cluster Assignments, NIPS 2020

Experiment

SelfAug improves the transferability of representations in various standard (first table) and few-shot (second table) downstream classification tasks

Method	CIFAR10	CIFAR100	Food	MIT67	Pets	Flowers	Caltech101	Cars	Aircraft	DTD	SUN397
<i>ImageNet100-pretrained ResNet-50</i>											
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

Method	FC100		CUB200		Plant Disease	
	(5, 1)	(5, 5)	(5, 1)	(5, 5)	(5, 1)	(5, 5)
<i>ImageNet100-pretrained ResNet-50</i>						
SimSiam	36.19 \pm 0.36	50.36 \pm 0.38	45.56 \pm 0.47	62.48 \pm 0.48	75.72 \pm 0.46	89.94 \pm 0.31
+ AugSelf (ours)	39.37\pm0.40	55.27\pm0.38	48.08\pm0.47	66.27\pm0.46	77.93\pm0.46	91.52\pm0.29
MoCo v2	31.67 \pm 0.33	43.88 \pm 0.38	41.67 \pm 0.47	56.92 \pm 0.47	65.73 \pm 0.49	84.98 \pm 0.36
+ AugSelf (ours)	35.02\pm0.36	48.77\pm0.39	44.17\pm0.48	57.35\pm0.48	71.80\pm0.47	87.81\pm0.33
Supervised	33.15 \pm 0.33	46.59 \pm 0.37	46.57 \pm 0.48	63.69 \pm 0.46	68.95 \pm 0.47	88.77 \pm 0.30
+ AugSelf (ours)	34.70\pm0.35	48.89\pm0.38	47.58\pm0.48	65.31\pm0.45	70.82\pm0.46	89.77\pm0.29

SelfAug can be incorporated with various SSL methods (STL10 pretraining)

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

Object localization (blue is ground-truth & red is prediction)

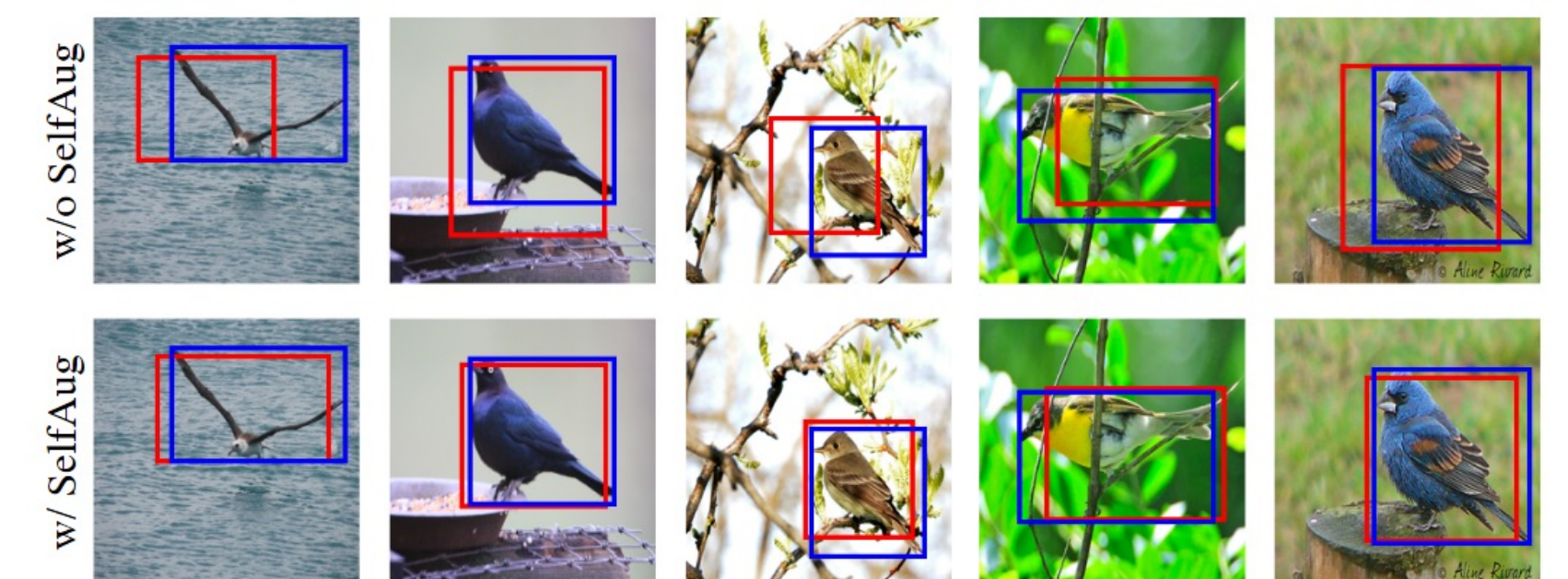


Image retrieval: SimSiam (left) vs SimSiam+AugSelf (right, ours)

