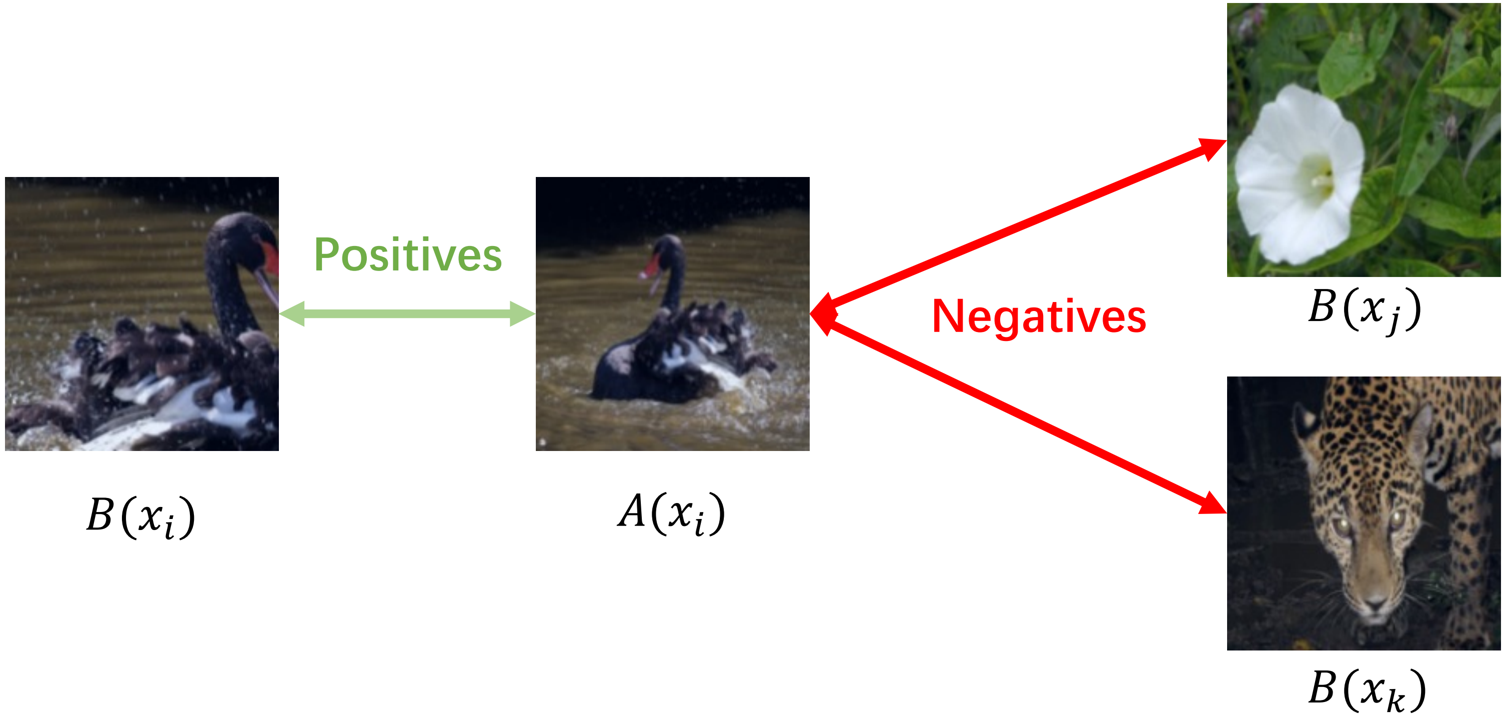
Identity-Disentangled Adversarial Augmentation for Self-Supervised Learning

Kaiwen Yang[1], Tianyi Zhou[2], Xinmei Tian[1,3], Dacheng Tao[4]

1. University of Science and Technology of China
2. University of Washington, Seattle; University of Maryland, College Park
3. Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
4. JD Explore Academy

Contrastive Learning (CL): Sample Identification Task



CL can be viewed as a sample identification task:

$$L_{\mathrm{NCE}}(\vec{x}) = -\frac{1}{N}\sum_{i=1}^{N} \log \boxed{q_{\mathrm{NCE}}(i|x = x_i)},$$

Identifying sample x_i or its augmentation as sample-i

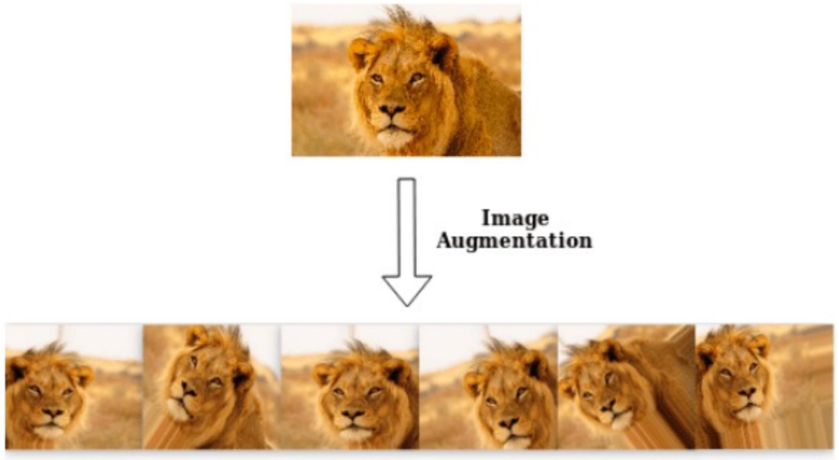
$$q_{\mathrm{NCE}}(i|x = x_i) \triangleq \frac{\exp\langle f(A(x_i)), h(B(x_i))\rangle}{\sum_{j=1}^{N} \exp\langle f(A(x_i)), h(B(x_j))\rangle},$$

Data Augmentation for Self-Supervised Learning

(1) Random Augmentation

-Uses pre-defined random image transformation.

-Carefully tune the hyperparameter for each transformation.



(2) Adversarial Augmentation

-CLAE [1] uses adversarial augmentation to generate hard positives/negatives.

[1] Chih-Hui Ho and Nuno Vasconcelos. Contrastive learning with adversarial examples. Neurips, 2020.

Data Augmentation for Self-Supervised Learning

(1) Random Augmentation
(Easy and identity-preserved):

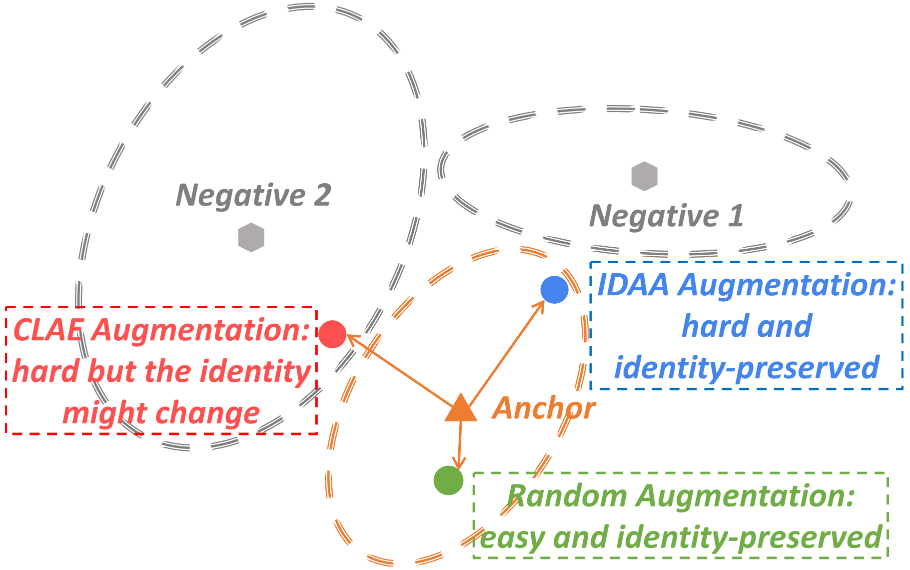
-Too easy for the sample identification task.

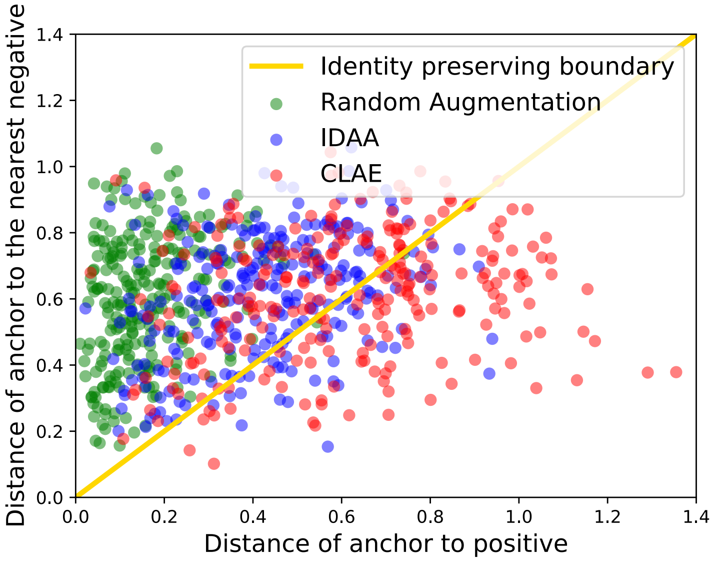
-Lead to nearly 0 loss and inefficient training.

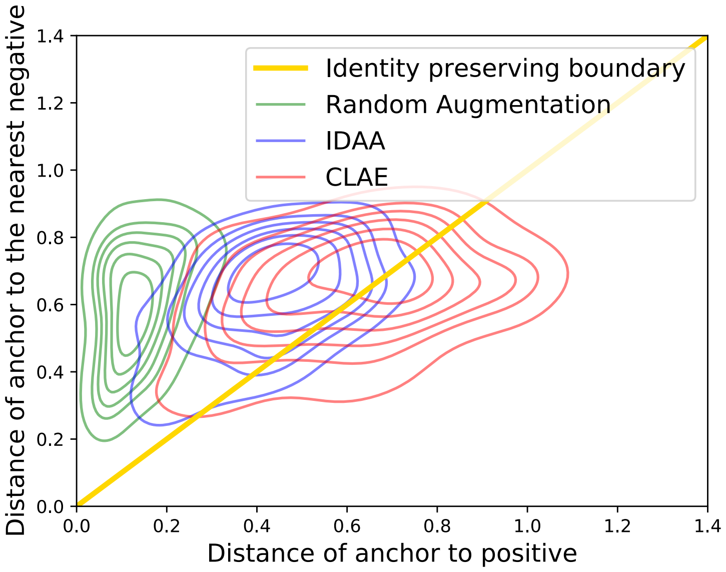
(2) Adversarial Augmentation
(Hard but the identity might change):

-May change the original sample identity.

-Infeasible to tune the attack strength for every sample to preserve the identity.







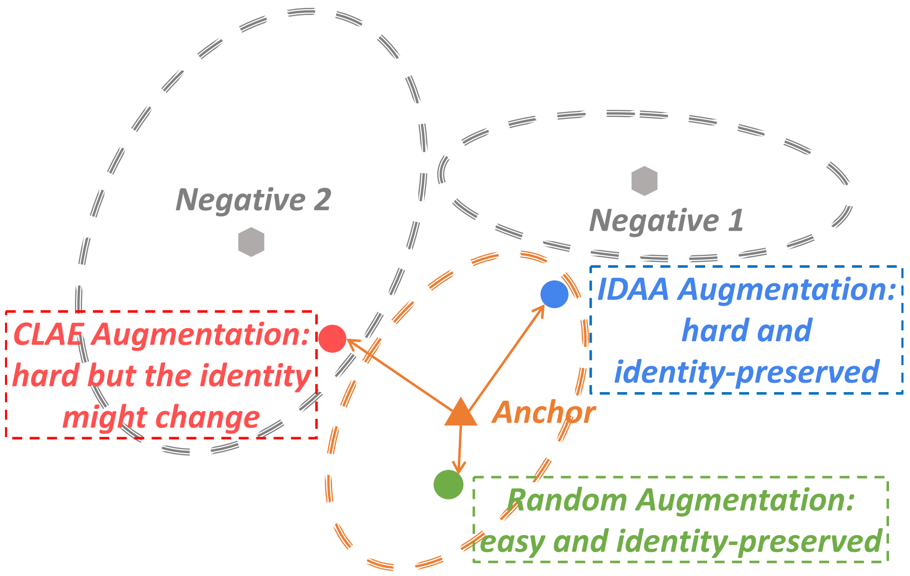
Data Augmentation for Self-Supervised Learning

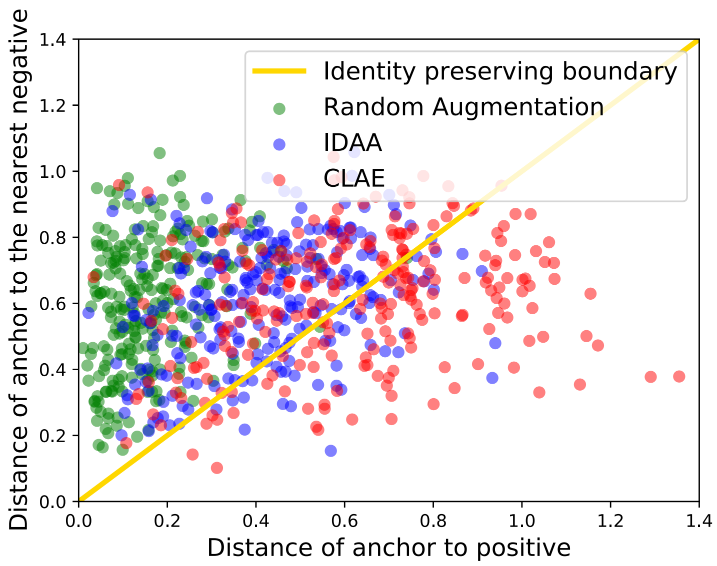
Aim: **(Hard and identity-preserved)** augmentation

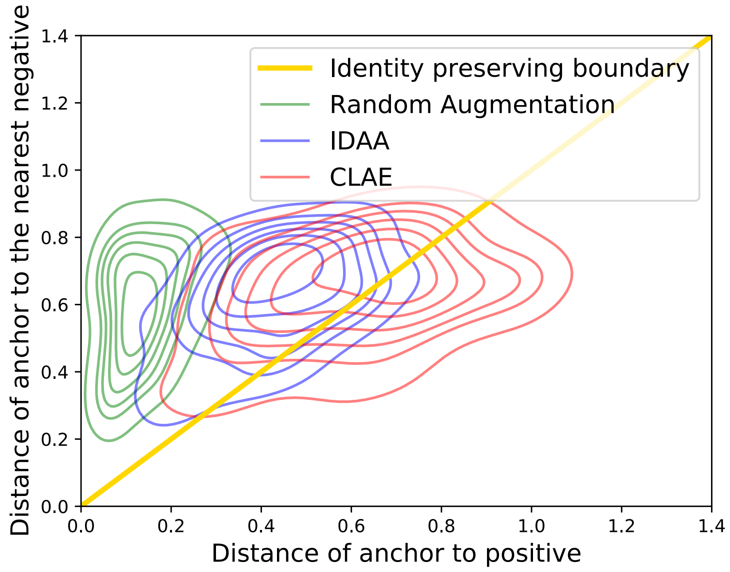
Main idea:

-**Disentangle** the sample into two parts: **identity-related** part and **identity-disentangled** part.

-Maintain the identity-related part **intact**, adversarailly **change** the identity-disentangled part.







Information-theoretic Interpretation

- **Identity-disentanglement via VAE**

Lemma 3.2. (VAE objective and $I(z;y)$ from Eq. (29) in (Alemi et al., 2016)). *Assume that the bottleneck features of VAE are denoted by z, the encoder is $E(\cdot)$ and produces distribution $p_E(z|x)$, the decoder is $D(\cdot)$ and produces distribution $q_D(x|z)$, the prior for z is $p(z)$, and the KL-divergence regularization in the VAE objective L_{VAE} has a weight β, we have:*

$$-I(z;x) + \boxed{\beta I(z;y)} \leq L_{\text{VAE}}, \tag{5}$$

$$L_{\text{VAE}} \triangleq -\int \mathrm{d}x p(x) \int \mathrm{d}z p_E(z|x) \log q_D(x|z) + \beta \frac{1}{N} \sum_{i=1}^{N} \mathrm{D}_{\mathrm{KL}} \left(p_E\left(z|x=x_i\right) || p(z)\right), \tag{6}$$

Identity-disentangled part: VAE reconstruction $G(x)$

Identity-related part: residual of VAE $R(x) \triangleq x - G(x)$

Information-theoretic Interpretation

Identity-disentangled data augmentation: $x' = \underline{R(x)} + \underline{G'(x)}$

Maintain the identity-related part $R(x)$ intact

change the identity-disentangled part $G(x)$ into $G'(x)$

- **Identity-preserving lower bound of the augmentation**

Theorem 3.5. (Identity-disentangled data augmentation). *If we use a VAE in the identity-disentangled data generative model for Lemma 3.3, and if we define an augmentation* $x' = R(x) + G'(x)$ *with* $G'(x) \sim q_D(x|z')$ *and* $z' = z + \delta$ *(a δ-perturbed z)), there exists a small* $\epsilon > 0$ *such that for any* $\|\delta\|_p \leq \epsilon$*, we can lower bound* $I(x'; y)$ *as*

$$I(x'; y) \geq I(x; y) - \boxed{\frac{1}{\beta}(L_{\mathrm{VAE}} + I(z; x))}. \quad (8)$$

Identity-disentangled Adversarial Augmentation (IDAA)

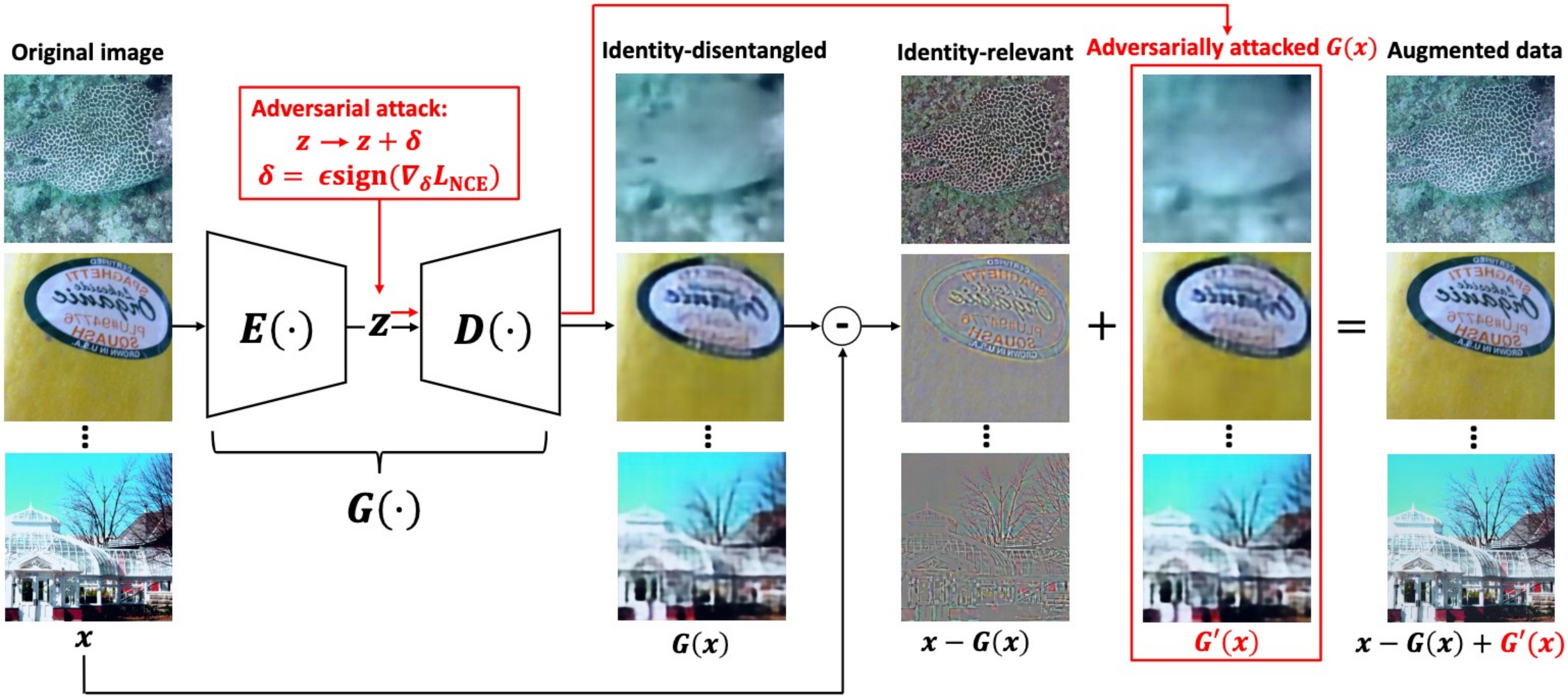


Figure 2. Architecture and pipeline of Identity-Disentangled Adversarial Augmentation (IDAA).

$$x' = R(x) + D(E(x) + \delta^*), \delta^* = \epsilon \mathrm{sign}(\nabla_\delta L_{\mathrm{NCE}}(\vec{x}'))$$

Experiments

- **Self-Supervised Learning Experiments**

IDAA brings **significant improvements** to many SSL methods (**both contrastive and non-contrastive methods**) on **mainstream benchmarks**, including CIFAR and ImageNet.

Method	kNN			Linear Evaluation		
	CIFAR10	CIFAR100	miniImageNet	CIFAR10	CIFAR100	miniImageNet
Plain	82.78±0.20	54.73±0.20	46.96±0.32	79.65±0.43	51.82±0.46	44.90±0.29
Plain+CLAE	83.09±0.19	55.28±0.12	47.01±0.28	79.94±0.28	52.14±0.21	45.43±0.15
Plain+IDAA	**86.00±0.16**	**58.64±0.15**	**47.83±0.29**	**82.83±0.10**	**56.12±0.16**	**46.81±0.16**
UEL	83.63±0.14	55.23±0.28	40.71±0.73	80.63±0.18	52.99±0.25	43.08±0.35
UEL+CLAE	84.00±0.15	55.96±0.06	41.75±0.39	80.94±0.13	54.27±0.40	44.32±0.24
UEL+IDAA	**86.69±0.13**	**59.04±0.18**	**43.24±0.32**	**83.65±0.17**	**57.25±0.19**	**45.74±0.30**
SimSiam	88.22±0.10	57.13±0.20	31.68±0.28	89.84±0.15	62.76±0.13	40.62±0.48
SimSiam+CLAE	85.59±0.21	53.88±0.08	27.77±3.47	87.77±0.08	60.89±0.22	37.32±0.47
SimSiam+IDAA	**89.08±0.12**	**58.19±0.19**	**32.14±0.58**	**90.99±0.18**	**65.21±0.37**	**41.24±0.51**
SimCLR	80.79±0.10	41.11±0.28	30.13±0.28	86.40±0.18	57.81±0.10	46.13±0.23
SimCLR+CLAE	80.27±0.18	43.57±0.17	32.23±0.08	85.25±0.07	57.69±0.25	46.76±0.16
SimCLR+IDAA	**83.41±0.22**	**46.78±0.22**	**33.66±0.16**	**88.07±0.22**	**60.90±0.08**	**48.23±0.23**

Method	Epoch	Batch Size	ImageNet Top-1	ImageNet Top-5
MoCo (He et al., 2020)	200	256	60.6	-
MoCo v2 (Chen et al., 2020b)	200	256	67.5	88.2
MoCHi (Kalantidis et al., 2020)	800	512	68.7	-
SimCLR (Chen et al., 2020a)	1000	4096	69.3	89.0
SwAV (Caron et al., 2020)	400	4096	70.1	-
AdCo (Hu et al., 2021)	200	256	68.6	-
InfoMin (Tian et al., 2020a)	200	256	70.1	89.4
SimSiam (Chen et al., 2020a)	100	256	68.1	-
SimSiam (Chen et al., 2020a)	200	256	70.0	-
SimSiam§	100	256	68.1	88.2
SimSiam§+IDAA	100	256	69.0	88.8
SimSiam§	200	256	69.8	89.2
SimSiam§+IDAA	200	256	**70.6**	**89.7**

Experiments

- **Transfer Learning Performance**

	CIFAR10	CIFAR100	Birdsnap	Aircraft	DTD	Pets	Flower	CUB-200
SimCLR	61.83	36.55	12.68	24.19	54.35	46.46	75.00	16.73
SimCLR+CLAE	61.59	37.13	13.61	25.87	52.12	43.55	76.82	17.58
SimCLR+IDAA	**64.49**	**38.82**	**13.89**	**26.02**	**54.97**	**46.76**	**77.99**	**18.15**

Method	COCO detection			COCO instance seg.		
	AP_{50}	AP	AP_{75}	AP_{50}^{mask}	AP^{mask}	AP_{75}^{mask}
scratch	44.0	26.4	27.8	46.9	29.3	30.8
ImageNet supervised	58.2	38.2	41.2	54.7	33.3	35.2
SimSiam (Chen et al., 2020a)	57.5	37.9	40.9	54.2	33.2	35.2
SimSiam+IDAA	**58.2**	**38.7**	**42.0**	**55.1**	**33.9**	**35.9**

- **Semi-Supervised Learning Performance**

Method	CIFAR100		
	400 labels	2500 labels	10000 labels
Fixmatch	47.76	66.30	74.13
Fixmatch+CLAE	50.34	68.58	74.54
Fixmatch+IDAA	**52.88**	**68.96**	**75.28**

Experiments

- **A Thorough Sensitivity Analysis**

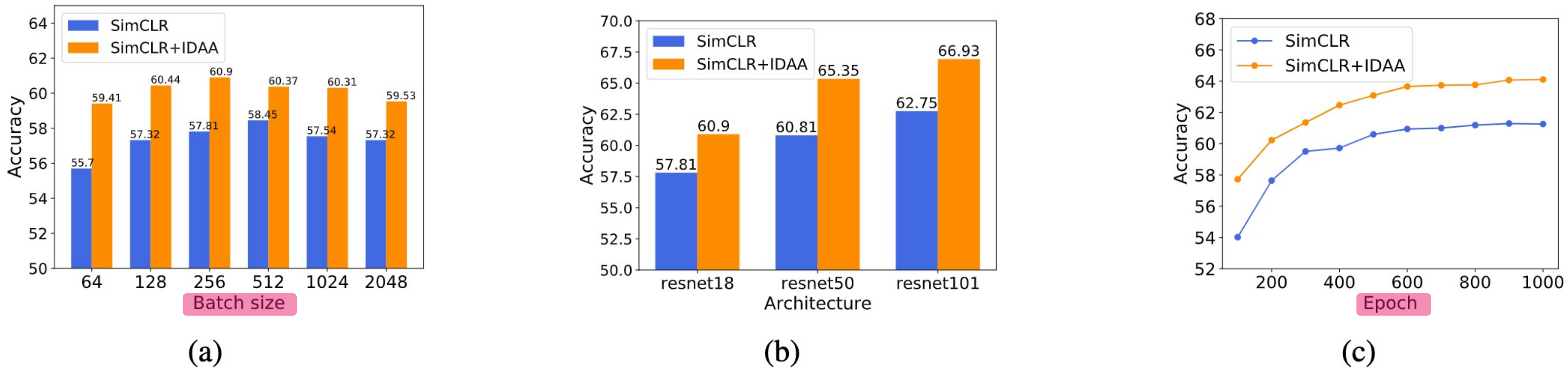


Figure 5. SSL performance under different (a) batch sizes, (b) ResNet architectures, and (c) training epochs.

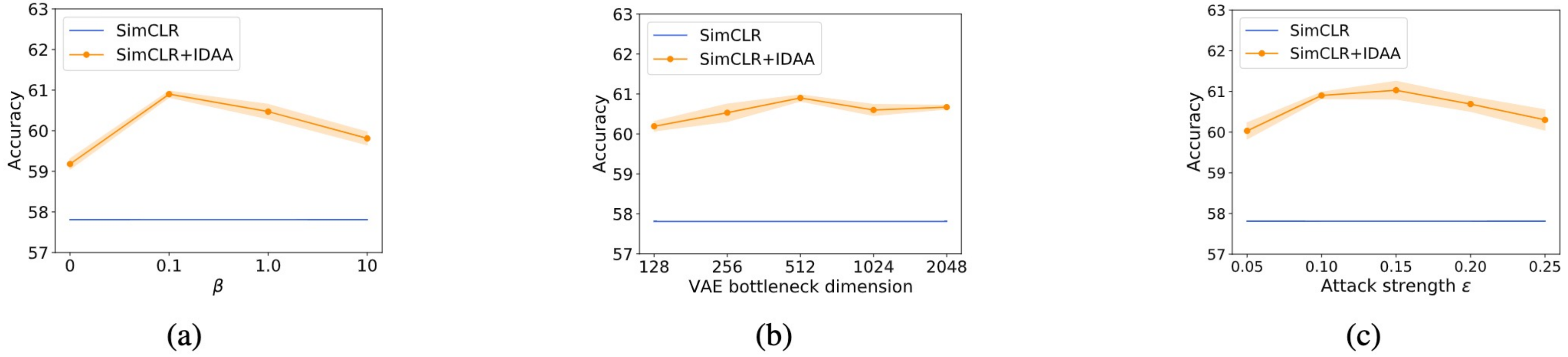


Figure 6. SSL performance using different (a) β, (b) VAE bottleneck dimensions, and (c) Attack strength ϵ.

Thanks!

Poster Session 3:
July 21st (Thursday) at 10:00 p.m-12:00 p.m UTC