DEEP NOID

Self-Supervised Learning by Estimating Twin Class Distributions

2023. 01. 18

Presenter : Taewoong Kang
twk@deepnoid.com

Abstract

- Self-Supervised Learning
- Classifying large-scale unlabeled datasets in end-to-end way (employ a Siamese network)
- Focused on enforcing the class distsribution of two different augmentations to be consistent
  - Simply minimizing divergence between augmentation
  - -> cause collapsed solution

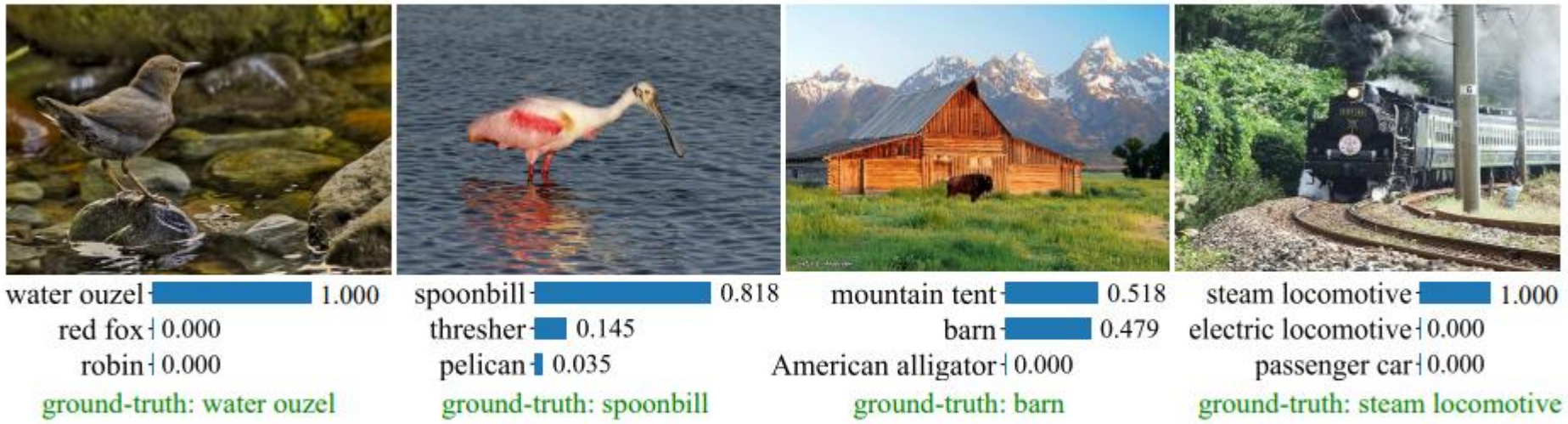


Figure 1. Examples of unsupervised top-3 predictions of TWIST. The predicted class indices are mapped to the labels in ImageNet by Kuhn–Munkres algorithm [34]. Note that the labels are only used to map our predictions to the ImageNet labels, we do not use any label to participate in the training process. More examples are given in Appendix.

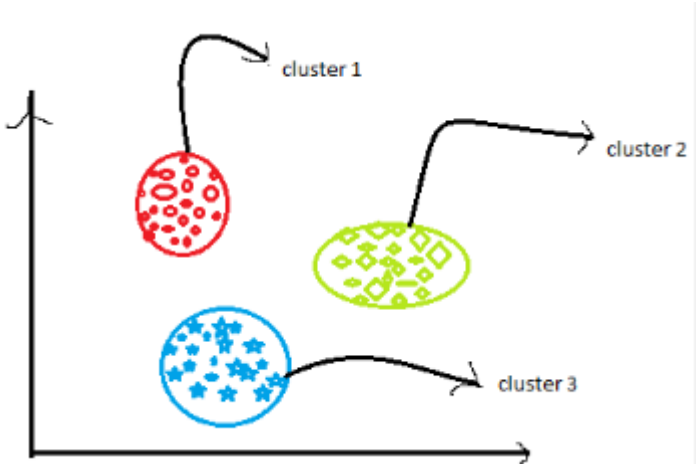
TWIST can avoid the collapsed solution

- Collapsed solution
  - Outputting the same class probability distributions for all images

- Maximize mutual information between the input and the class prediction
  - Maximizing the agreement of different augmentation
  - Pushing away the representation of different image based on the instance discrimination pretext task

Contrastive method

- Learn an embedding space
  - Features of different augmentation from the same images are attracted
  - Features of different images are separated



Contrastive method

- BYOL, SimSiam
  - Abandon the negative samples and design some special techniques
  - Such as asymmetric architecture, momentum encoder and stop gradients
- Barlow Twins, VICReg
  - Learn informative representations
  - By reducing the redundancy or covariance of different dimensions

Clustering-based method

- Use clustering tool
- To generate pseudo-labels for images and classify the images with the generated pseudo-labels

- DeepCluster
  - K-means
- SwAV
  - Sinkhorn-Knopp
- DINO
  - Updates pseudo-targets using the output of the momentum teacher together with sharpening and centering operations
- Self-Classifier
  - End-to-end method to classify unlabeled datasets
- SCAN
  - Two-step approach to mainly focus on unsupervised classification task

Mutual information maximization

- Bridle, IMSAT, IIC, Deep InfoMax
- Difference between Deep InfoMax and TWIST
  - The task to classify image by exploring their semantric relation
  - Information between images and a discrete random variable, instead of high-dim continuous representation
  - Do not require the neural estimator

Architecture

- Siamese Network
- X1, X2 are two augmented version of X
- The outputs are two probability distribution over C categories

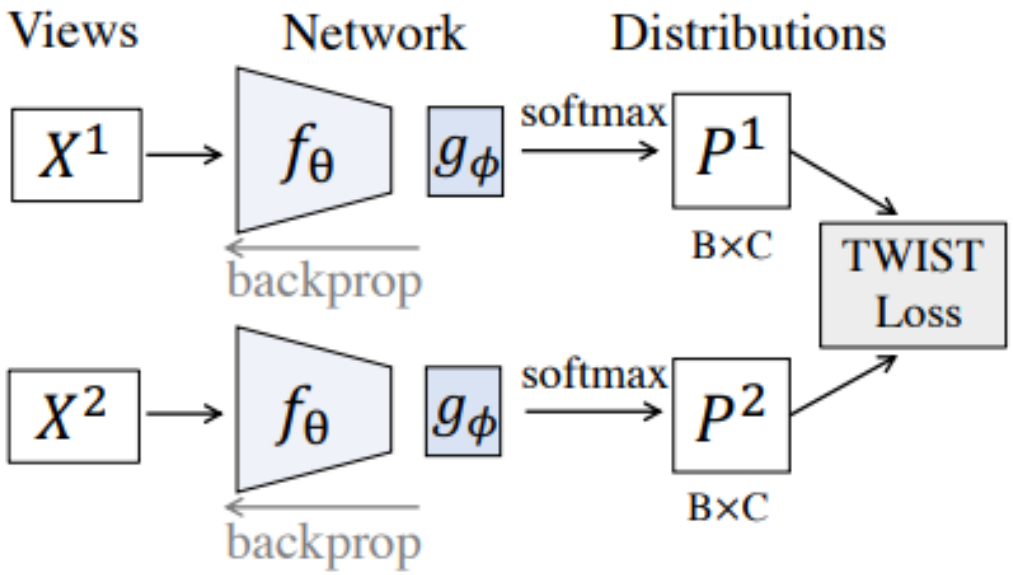


Figure 2. Network architecture of TWIST.

Learning objective

- Symmetric form

- H -〉 Entropy
- α, β are hyper-parameters
- D_{KL} -〉 Kullback-Leibler divergence
  - Relative Entropy
  - Cross entropy - entropy
  - P1, P2가 가까워 지게

$$\mathcal{L}(P^1, P^2) = \underbrace{\frac{1}{2B}\sum_{i=1}^{B}(D_{KL}(P_i^1||P_i^2) + D_{KL}(P_i^2||P_i^1))}_{\text{consistency term}} + \frac{\alpha}{2}\sum_{k=1}^{2}\underbrace{\frac{1}{B}\sum_{i=1}^{B}H(P_i^k)}_{\text{sharpness term}} - \frac{\beta}{2}\sum_{k=1}^{2}\underbrace{H(\frac{1}{B}\sum_{i=1}^{B}P_i^k)}_{\text{diversity term}},$$

- $D_{KL}(P_i^1||P_i^2) = H(P_i^1, P_i^2) - H(P_i^1)$

$$= -\sum_i P_i^1 \log \frac{P_i^2}{P_i^1}$$

DEEP NOID

Learning objective

- Sharpening term
  - Minimize the entropy of class distribution for each samples to regularize the output distribution to be sharp
  - Makes each sample have a deterministic assignment
- Diversity term
  - Make the predictions for different samples be diversely distributed
  - Maximizing the entropy of the mean distribution across different samples

$$\mathcal{L}(P^1, P^2) = \underbrace{\frac{1}{2B}\sum_{i=1}^{B}(D_{KL}(P_i^1||P_i^2) + D_{KL}(P_i^2||P_i^1))}_{\text{consistency term}}$$

$$+ \frac{\alpha}{2}\sum_{k=1}^{2}\underbrace{\frac{1}{B}\sum_{i=1}^{B}H(P_i^k)}_{\text{sharpness term}} - \frac{\beta}{2}\sum_{k=1}^{2}\underbrace{H(\frac{1}{B}\sum_{i=1}^{B}P_i^k)}_{\text{diversity term}},$$

Theoretical Explanation

- Mutual Information ≈ Sharpening term + Diversity term
  - Derived from the Monte Carlo estimation

$$
\begin{aligned}
-I(X,Y) &= H(Y|X) - H(Y) \\
&= \mathop{\mathbb{E}}_{x}\left[-\sum_{y} p(y|x)\log p(y|x)\right] - H(Y) \\
&\approx \underbrace{-\frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}\sum_{y} p(y|x)\log p(y|x)}_{\text{sharpness term}} \underbrace{- H\Big(\frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} p(Y|x)\Big)}_{\text{diversity term}},
\end{aligned}
$$

$$
p(Y) = \int_{x} p(Y|x)p(x)dx \approx \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} p(Y|x).
$$

Ablation Study on loss terms

L_s	L_d	$L_s =$	$L_d =$	$\|g\|$	acc
✗	✗	8.28	8.28	0	0.1
✗	✓	8.27	8.28	0	0.1
✓	✗	2.59	6.42	0.01	56.1
✓	✓	1.51	7.87	0.02	70.9

Table 8. Ablation study on the loss terms. Here L_s and L_d denote the sharpness and diversity term respectively. $|g|$ denotes the mean magnitude of gradients before the last batch normalization and "acc" is the linear accuracy. Models are trained for 50 epochs.

- Without sharpness term generate collapsed solutions
- Without diversity term do not generate collapsed solution
  - But performances deteriorate significantly
  - Theoretically, will lead to collapsed solution
  - But the NBS helps avoid the problem
  - (can separate the probabilities in different column and force them to have a unit standard deviation)

Amplifying Variance for Better Optimization

- Consistency, sharpness term are easy to minimize
- But diversity term is difficult to maximize
- Column Standard Deviation keeps small -〉 cause low diversity
- To solve this problem -〉 add a batch normalization before the softmax to amplify the variance to force them to be separated =〉5% improvement in ImageNet linear Classification

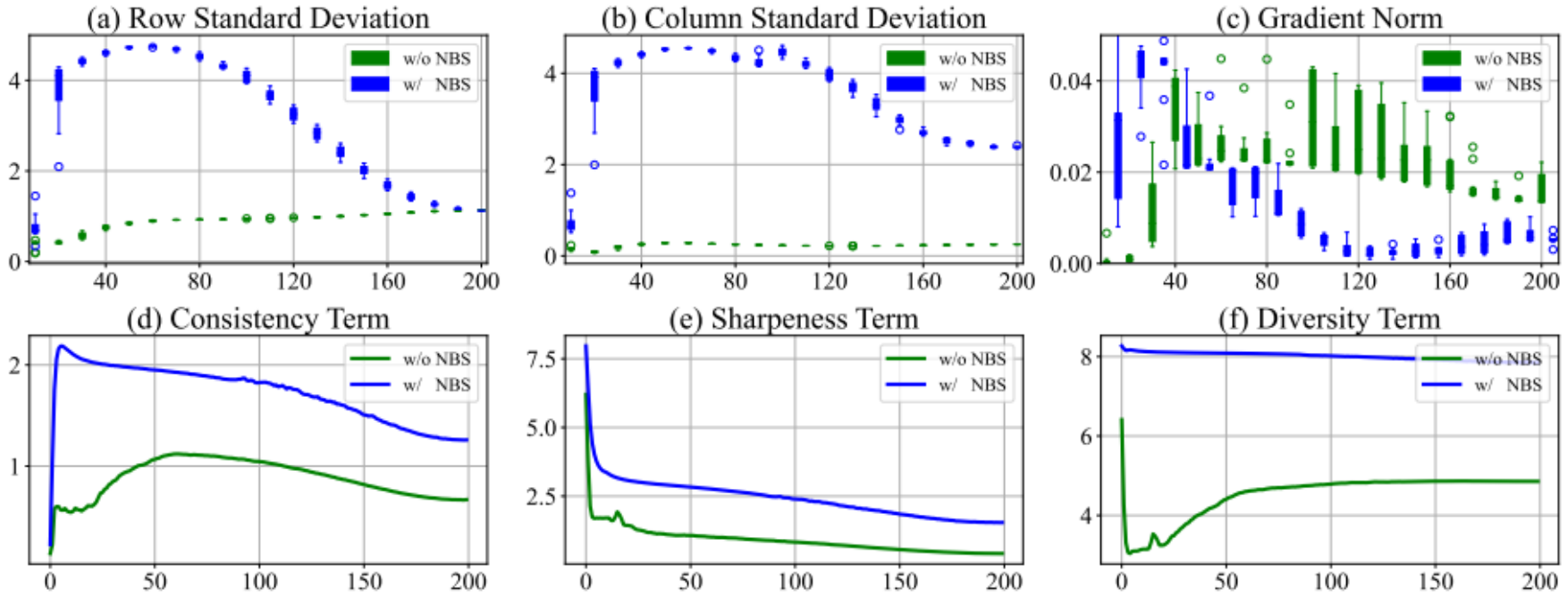


Figure 3. We show the statistical characteristics of the output before softmax operation with and without NBS and the training curves.

Ablation Study on batch normalization before softmax

NBS	Loss	ACC	NMI	std_c	std_r
✓	-5.05	70.6	59.0	2.37	1.12
✗	-3.78	65.5	47.0	0.26	1.14

Table 7. Ablation study on batch normalization before softmax.

- NBS (batch normalization before softmax)

Self-labeling for ResNets

- The multi-crop strategy gaves the performance improvement of TWIST is much smaller than SwAV.
- SwAV uses the global crops to generate accurate pseudo-labels as supervision to train the local crop
- In TWIST, global and local crops are regarded same
- => add a self-labeling stage after the regular training

Method	Network	Param	Epoch	Top1	Top5
ResNet-50 without multi-crop					
MoCo v2	RN50	24M	800	71.1	90.1
SimCLR	RN50	24M	1000	69.3	89.0
BarlowTwins	RN50	24M	1000	73.2	91.0
BYOL	RN50	24M	1000	**74.3**	**91.6**
SelfClassifier	RN50	24M	800	69.7	89.3
SwAV	RN50	24M	800	71.8	-
TWIST	RN50	24M	800	72.6	91.0
ResNet-50 with multi-crop					
SwAV	RN50	24M	800	75.3	-
DINO	RN50	24M	800	75.3	92.5
TWIST	RN50	24M	300	75.0	92.4
TWIST	RN50	24M	800	**75.5**	**92.5**

Table 3. Linear classification results. We report and compare results with different backbones.

Ablation Study on Multi-crop and Self-labeling

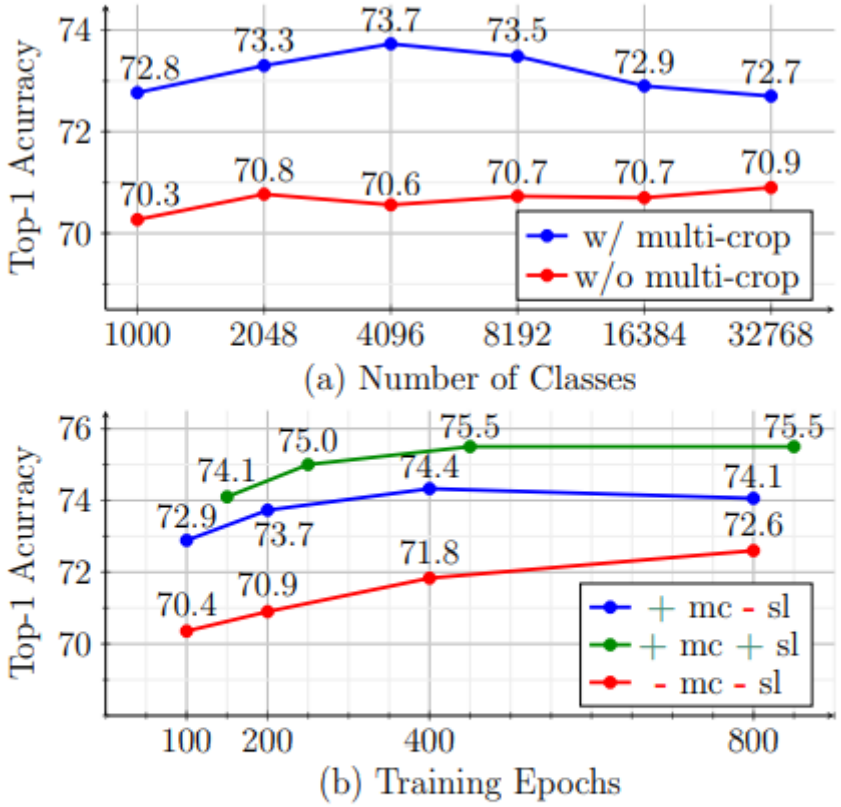


Figure 4. **(a)** Effect of different numbers of classes in TWIST. **(b)** Effect of different training epochs in TWIST, where "mc" denotes multi-crop and "sl" denotes self-labeling. All results are ImageNet one-crop top-1 accuracy.

Momentum Encoder for ViT

- Adopt the momentum encoder design, which is widely adopted to train ViT-based models.
- Although TWIST using ViT as backbone without momentum encoder can work well, we use it only for accuracy gains

DEEP NOID

Semi/Full-supervised Fine-tuning

- Fine-tuned the pre-trained TWIST model on a subset of ImageNet
- TWIST outperforms all other SOTA method by large margins

Method	1% Labels		10% Labels		100% Labels	
	Top1	Top5	Top1	Top5	Top1	Top5
ResNet-50						
SUP	25.4	48.4	56.4	80.4	76.5	-
SimCLR	48.3	75.5	65.6	87.8	76.5	93.5
BYOL	53.2	78.4	68.8	89.0	77.7	93.9
SwAV	53.9	78.5	70.2	89.9	-	-
DINO	52.2	78.2	68.2	89.1	-	-
BarlowTwins	55.0	79.2	69.7	89.3	-	-
TWIST	**61.2**	**84.2**	**71.7**	**91.0**	**78.4**	**94.6**
ResNet-50×2						
SimCLR	58.5	83.0	71.7	91.2	-	-
BYOL	62.2	84.1	73.5	91.7	-	-
TWIST	**67.2**	**88.2**	**75.3**	**92.8**	**80.3**	**95.4**
ViT-B/16						
DINO	67.3	88.2	74.6	92.0	**82.8**	-
TWIST	**69.6**	**89.7**	**76.5**	**93.1**	**82.8**	**96.3**

Table 1. Semi-supervised classification results on ImageNet. We report top-1 and top-5 center-crop accuracies, from 1% to 100%.

Unsupervised classification

Method	NMI	ARI	AMI	ACC
SCAN	72.0	27.5	51.2	39.9
SeLa	65.7	16.2	42.0	-
SelfClassifier	64.7	13.2	46.2	-
TWIST	**74.3**	**30.0**	**57.7**	**40.6**

Table 2. Unsupervised classification results on ImageNet. All numbers are reported on the validation set of ImageNet. Comparison methods include SCAN [53], SeLa [2], and Self Classifier [1].

- The outputs are directly mapped to the real labels by the Kuhn-Munkres algorithm

DEEP NOID

Evaluation Metrics

- 1.NMI(normalized mutual information)
  - problem : does not penalize large cardinalities
  - (i.e., over clustering)
- 2. ARI(adjusted rand index)
  - The corrected-for-chance version of the Rand index
  - RI(rand index) quantifies the percentage of correct decision
- 3. ACC(accuracy)
  - The standard measure
- 4. AMI(adjusted mutual information)
- 5. Sillhouette Score
  - The more data points within each cluster are closely-packed and different clusters are well-separated

$$\mathrm{NMI} = \frac{2 \times I(\boldsymbol{y}; \boldsymbol{z})}{H(\boldsymbol{y}) + H(\boldsymbol{z})}$$

$$ARI = \frac{\sum_{kl}\binom{n_{kl}}{2} - [\sum_k \binom{a_k}{2} \sum_l \binom{b_l}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_k \binom{a_k}{2} + \sum_l \binom{b_l}{2}] - [\sum_k \binom{a_k}{2} \sum_l \binom{b_l}{2}]/\binom{n}{2}}$$

$$\mathrm{ACC} = \max_m \left(\frac{\sum_{i=1}^{N} \mathbb{1}(y_i = m(z_i))}{N} \right)$$

$$AMI(U,V) = \frac{MI(U,V) - E\{MI(U,V)\}}{\max\{H(U), H(V)\} - E\{MI(U,V)\}}.$$

$$s(i) = \begin{cases} \frac{b(i)-a(i)}{\max(a(i),b(i))}, & \text{if } N_k > 1 \\ 0, & \text{otherwise} \end{cases}$$

$$a(i) = \frac{1}{|N_k| - 1} \sum_{\boldsymbol{x}_j : z_j = k} d(\boldsymbol{x}_i, \boldsymbol{x}_j) \qquad b(i) = \min_{k' : k' \neq k} \frac{1}{N_{k'}} \sum_{\boldsymbol{x}_j : z_j = k'} d(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

DEEP NOID

Linear classification

Method	Network	Param	Epoch	Top1	Top5
ResNet-50 without multi-crop					
MoCo v2	RN50	24M	800	71.1	90.1
SimCLR	RN50	24M	1000	69.3	89.0
BarlowTwins	RN50	24M	1000	73.2	91.0
BYOL	RN50	24M	1000	**74.3**	**91.6**
SelfClassifier	RN50	24M	800	69.7	89.3
SwAV	RN50	24M	800	71.8	-
TWIST	RN50	24M	800	72.6	91.0
ResNet-50 with multi-crop					
SwAV	RN50	24M	800	75.3	-
DINO	RN50	24M	800	75.3	92.5
TWIST	RN50	24M	300	75.0	92.4
TWIST	RN50	24M	800	**75.5**	**92.5**
Wider ResNet					
SimCLR	RN50w2	94M	1000	74.2	92.0
CMC	RN50w2	94M	-	70.6	89.7
SwAV	RN50w2	94M	800	77.3	-
BYOL	RN50w2	94M	1000	77.4	93.6
TWIST	RN50w2	94M	300	**77.7**	**93.9**
Vision Transformer					
MoCo-v3	Deit-S/16	21M	300	72.5	-
DINO	Deit-S/16	21M	300	75.9	-
TWIST	Deit-S/16	21M	300	**76.3**	**92.7**
MoCo-v3	ViT-B/16	86M	300	76.5	-
DINO	ViT-B/16	86M	800	78.2	**93.9**
TWIST	ViT-B/16	86M	300	**78.4**	93.8

Table 3. Linear classification results. We report and compare results with different backbones.

- TWIST outperforms other SOTA method
- For wider ResNet50, also works well
- ViT is very sensitive to hyper-parameters and training it is very costly

- The performance of linear classification on ImageNet

Transfer Learning

Method	Food	Cifar10	Cifar100	Sun397	Cars	Aircraft	VOC	DTD	Pets	Caltech	Flowers	Avg
Linear evaluation:												
SimCLR	68.4	90.6	71.6	58.8	50.3	50.3	80.5	74.5	83.6	90.3	91.2	73.6
BYOL	75.3	91.3	**78.4**	62.2	**67.8**	60.6	82.5	75.5	90.4	94.2	**96.1**	**79.5**
SUP	72.3	**93.6**	78.3	61.9	66.7	**61.0**	82.8	74.9	91.5	**94.5**	94.7	79.3
TWIST	**78.0**	91.2	74.4	**66.8**	55.2	53.6	**85.7**	**76.6**	**91.6**	91.1	93.4	78.0
Fine-tune:												
Random	86.9	95.9	80.2	53.6	91.4	85.9	67.3	64.8	81.5	72.6	92.0	79.3
SimCLR	87.5	97.4	85.3	63.9	91.4	87.6	84.5	75.4	89.4	91.7	96.6	86.4
BYOL	88.5	97.8	86.1	63.7	91.6	**88.1**	85.4	76.2	91.7	**93.8**	97.0	87.3
SUP	88.3	97.5	86.4	64.3	**92.1**	86.0	85.0	74.6	92.1	93.3	**97.6**	87.0
TWIST	**89.3**	**97.9**	**86.5**	**67.4**	91.9	85.7	**86.5**	**76.4**	**94.5**	93.5	97.1	**87.9**

Table 4. Transfer learning results on eleven datasets, including linear evaluation and fine-tuneing. We use ResNet-50 as backbone and pre-trained on ImageNet. We calculate the average of performanes on these datasets and report it at the last column. TWIST performs best on the fine-tuning setting, which is in accordance to the advantage on the semi-supervised fine-tuning setting.

- The outputs are directly mapped to the real labels by the Kuhn-Munkres algorithm

DEEP NOID

Detection and Segmentation

Method	VOC07+12 detection			COCO detection			COCO instance seg		
	AP_{all}	AP_{50}	AP_{75}	AP^{bb}_{all}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}_{all}	AP^{mk}_{50}	AP^{mk}_{75}
Moco-v2	56.4	81.6	62.4	39.8	59.8	43.6	36.1	56.9	38.7
SimCLR †	**58.2**	83.8	65.1	41.6	61.8	45.6	37.6	59.0	40.5
SwAV	57.2	83.5	64.5	41.6	62.3	**45.7**	**37.9**	59.3	**40.8**
DC-v2 †	57.0	83.7	64.1	41.0	61.8	45.1	37.3	58.7	39.9
DINO †	57.2	83.5	63.7	41.4	62.2	45.3	37.5	58.8	40.2
DenseCL	56.9	82.0	63.0	40.3	59.9	44.3	36.4	57.0	39.2
TWIST	58.1	**84.2**	**65.4**	**41.9**	**62.6**	**45.7**	**37.9**	**59.7**	40.6

Table 5. Detection and instance segmentation. †means that we download the pre-trained models and conduct the experiments. For the VOC dataset, we run five trials and report the average. The performance is measured by Average Precision (AP). DC-v2 denotes the DeepCluster-v2.

Method	FCN-FPN	
	VOC	Cityscapes
Sup	67.7	**75.4**
Moco-v2	67.5	**75.4**
SimCLR	72.8	74.9
SwAV	71.9	74.4
DC-v2	72.1	73.8
DINO	71.9	73.8
TWIST	**73.3**	74.6

Table 6. Results of semantic segmentation with FCN-FPN backbone. All results are averaged over five trials.

- Use ResNet-50 with FPN
- For Pascal VOC, use Faster R-CNN
- For MSCOCO, use Mask R-CNN
- In implementation, use Detectron2

- Using FCN as architecture
- Use MMSegmentaion to train the architectures

Conclusion

- TWIST is simple and theoretically explainable
- Does not rely on any clustering tools

Limitation

- Vulnerable to data distribution (tries to capture the intrinsic semantic structure)
- With biased dataset, the model likely to learn malicious information
- When using ViT, adopt momentum encoder
- Performs best when the number of classes is 4096 (with Multi-crop)

DEEP NOID

Q & A

DEEP NOID

감사합니다