Self-Supervised Learning

Chaeeun Ryu, Jian Zhang,
Niranjan Sundararajan, Yousif Alozairi

Paper choice

Jian Zhang: 4.0, 4.4, 4.6

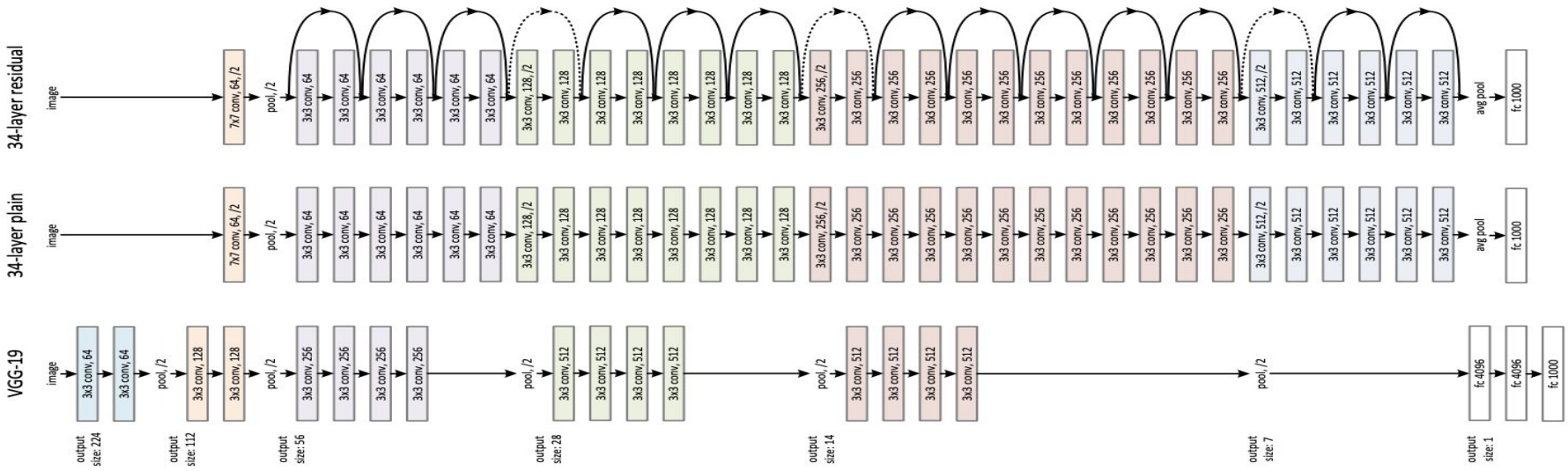
Yousif Alozairi: 4.3, 4.10

Chaeeun Ryu: 4.1, 4.2

Niranjan Sundararajan: 4.7, 4.8

Why?

Model gets larger, hard to train, requires larger dataset…



What is self-supervised learning?

- A form of unsupervised learning where the data provides the supervision through a proxy task.

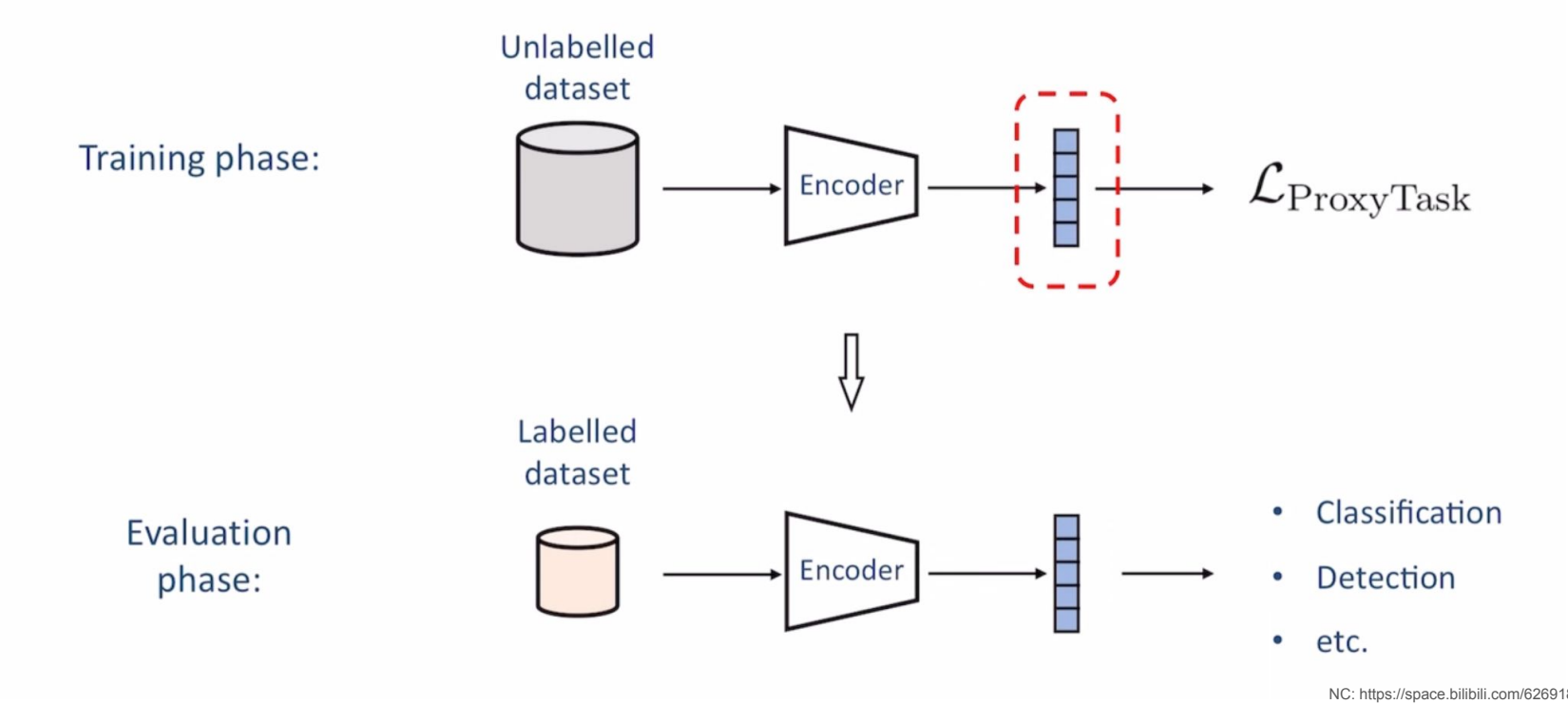
- Two different paradigms:

  • Define a proxy task, in order to solve it, the network is forced to learn what we really care about, e.g. a semantic representation. Evaluate the representation on downstream task with linear probing, e.g. SimCLR, MOCO, etc. (learning transferable features)

  • Define a proxy task, the network is directly generalizable to downstream tasks by solving the proxy one. Zero annotation is further required, e.g. audio-visual localization, tracking. (learning applicable embeddings)

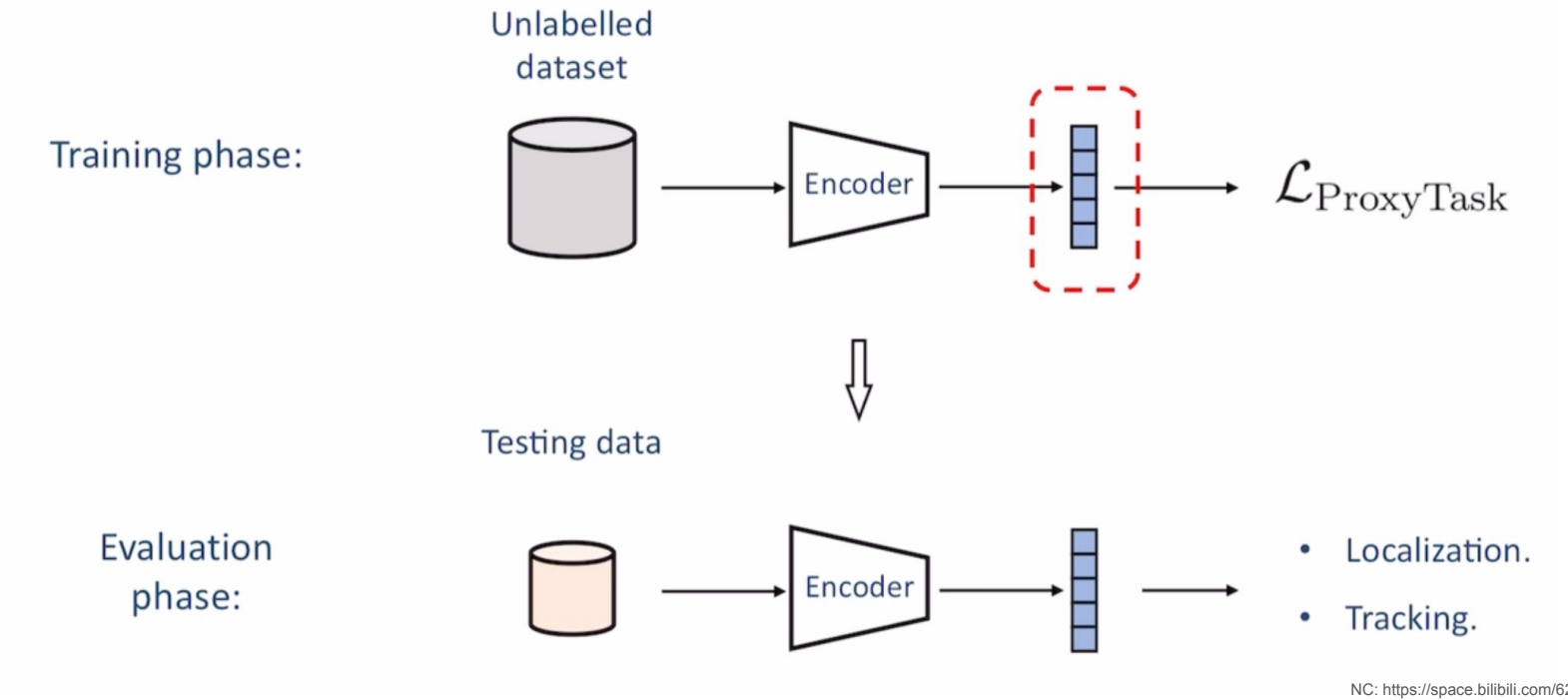
What is self-supervised learning?

Paradigm 1:



What is self-supervised learning?

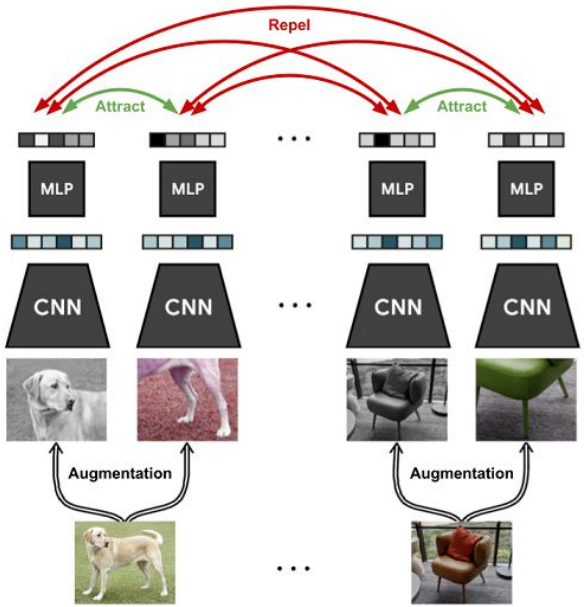
Paradigm 2:



Paper 4.1 & 4.2

Placeholder for Chaeeun Ryu

A Simple Framework for Contrastive Learning of Visual Representations

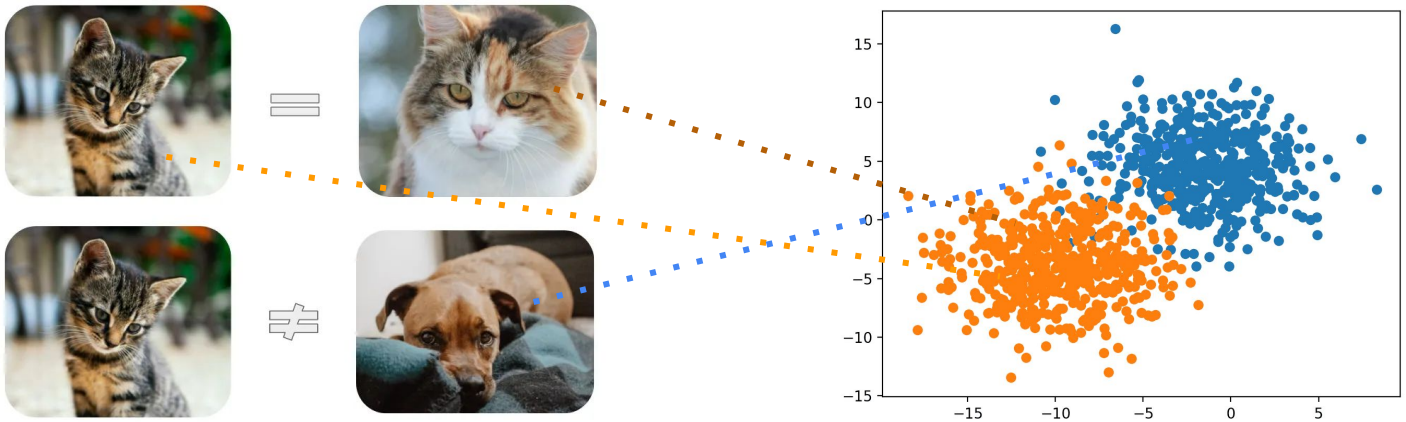


Ref: https://docs.google.com/presentation/d/1ccddJFD_j3p3h0TCqSV9ajSi2y1yOfh0-lJoK29ircs/edit#slide=id.g88de1f3f96_5_729

What is Contrastive Learning?

Core Idea:

1. Creating pairs of similar (positive) and dissimilar (negative) examples.
2. Training a model to bring similar examples closer together in a "feature space" while pushing dissimilar examples farther apart



Algorithm

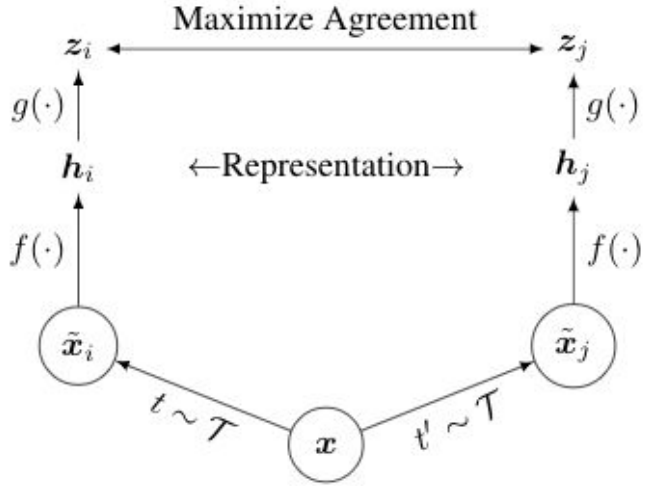


Figure 2. A framework for contrastive representation learning. Two separate stochastic data augmentations $t, t' \sim \mathcal{T}$ are applied to each example to obtain two correlated views. A base encoder network $f(\cdot)$ with a projection head $g(\cdot)$ is trained to maximize agreement in *latent representations* via a contrastive loss.

Algorithm

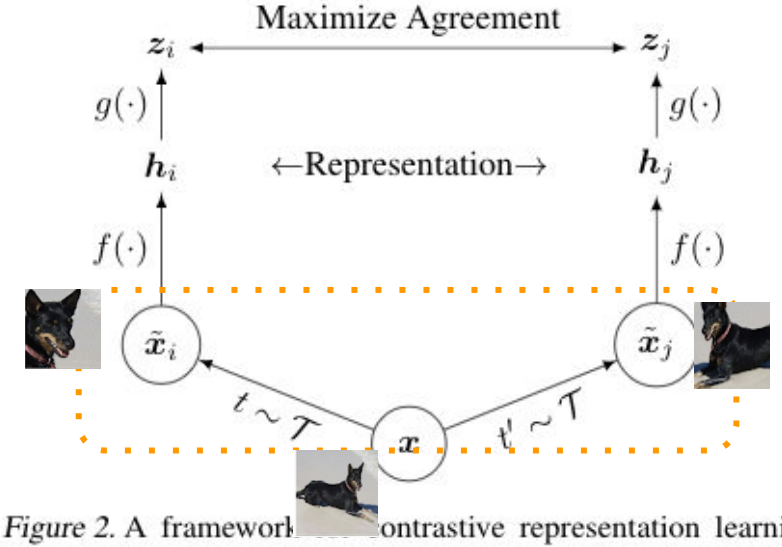
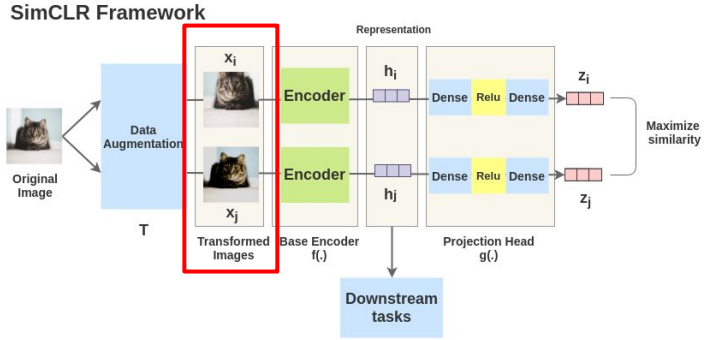


Figure 2. A framework [illegible]ontrastive representation learning. Two separate stochastic data augmentations $t, t' \sim \mathcal{T}$ are applied to each example to obtain two correlated views. A base encoder network $f(\cdot)$ with a projection head $g(\cdot)$ is trained to maximize agreement in *latent representations* via a contrastive loss.



Used random crop and color distortion for augmentation.

Algorithm

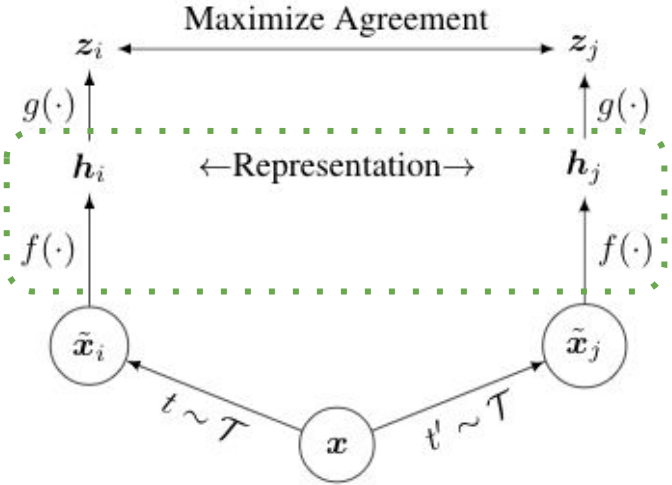
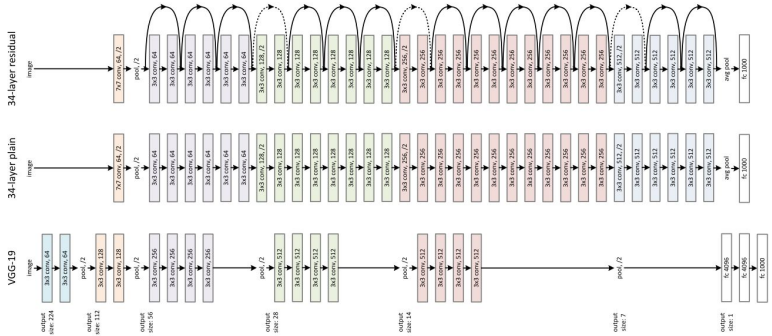


Figure 2. A framework for contrastive representation learning. Two separate stochastic data augmentations $t, t' \sim \mathcal{T}$ are applied to each example to obtain two correlated views. A base encoder network $f(\cdot)$ with a projection head $g(\cdot)$ is trained to maximize agreement in *latent representations* via a contrastive loss.

f(x) is the base network (encoder) that computes internal representation.

We use (unconstrained) ResNet in this work. However, it can be other networks.



Algorithm

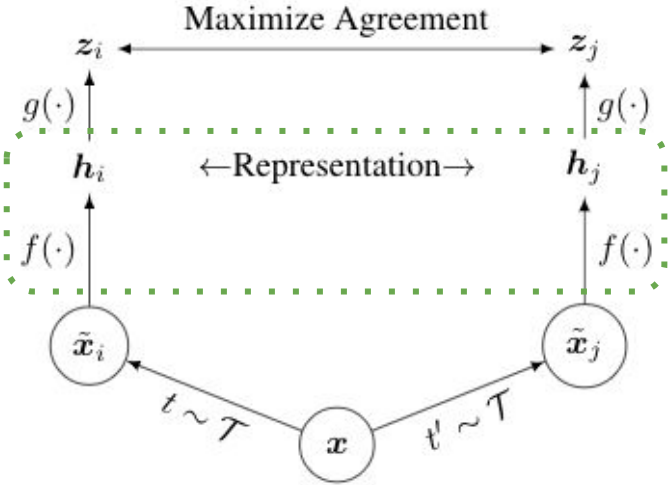
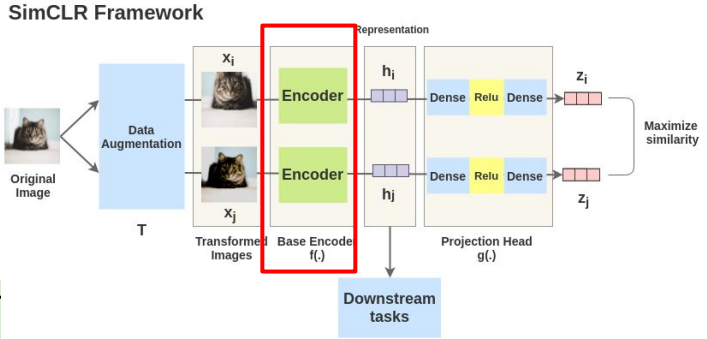
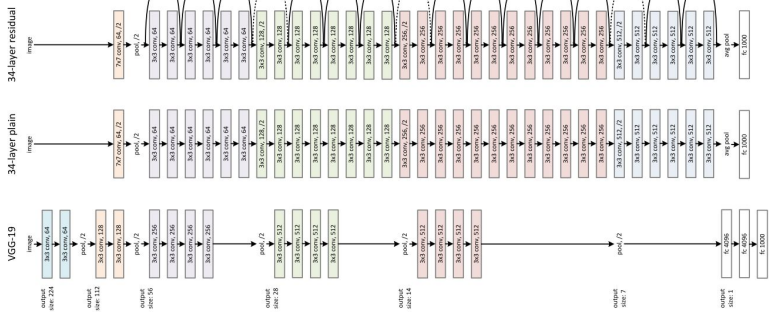


Figure 2. A framework for contrastive representation learning. Two separate stochastic data augmentations $t, t' \sim \mathcal{T}$ are applied to each example to obtain two correlated views. A base encoder network $f(\cdot)$ with a projection head $g(\cdot)$ is trained to maximize agreement in *latent representations* via a contrastive loss.



f(x) is th … nternal representation.

We use (unconstrained) ResNet in this work. However, it can be other networks.



Algorithm

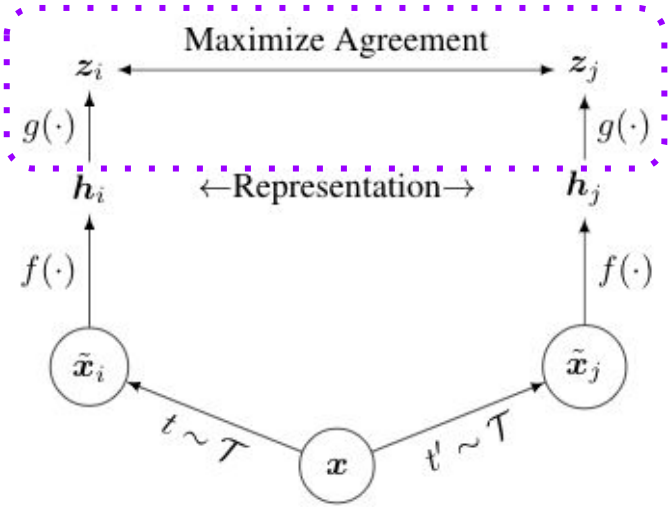
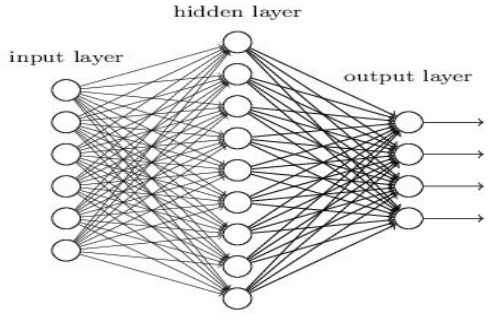


Figure 2. A framework for contrastive representation learning. Two separate stochastic data augmentations $t, t' \sim \mathcal{T}$ are applied to each example to obtain two correlated views. A base encoder network $f(\cdot)$ with a projection head $g(\cdot)$ is trained to maximize agreement in *latent representations* via a contrastive loss.

g(h) is a projection network that project representation to a latent space.

We use a 2-layer non-linear MLP (fully connected net).



Algorithm

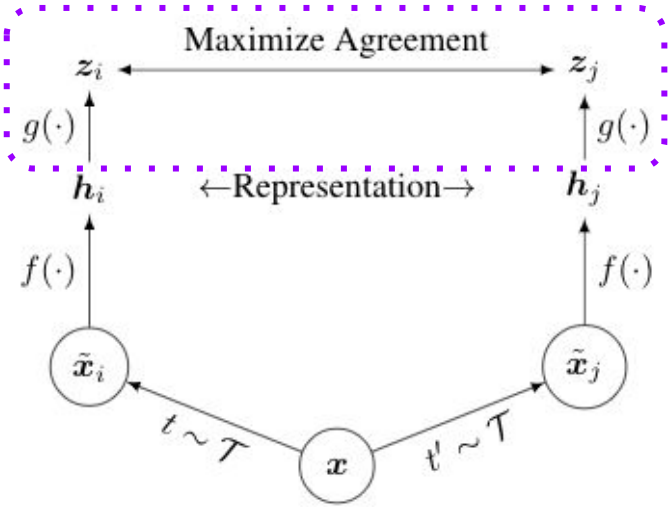
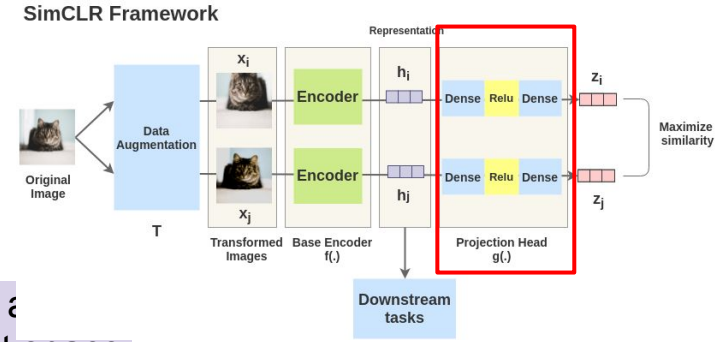
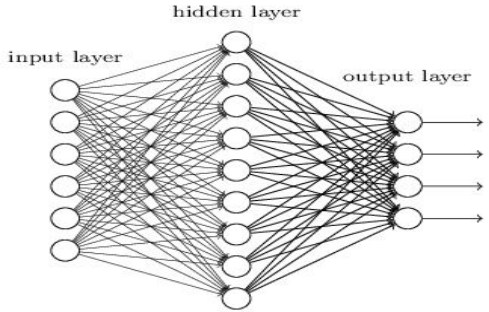


Figure 2. A framework for contrastive representation learning. Two separate stochastic data augmentations $t, t' \sim \mathcal{T}$ are applied to each example to obtain two correlated views. A base encoder network $f(\cdot)$ with a projection head $g(\cdot)$ is trained to maximize agreement in *latent representations* via a contrastive loss.



g(h) is a tion to a latent space.

We use a 2-layer non-linear MLP (fully connected net).



Algorithm: Loss

for all $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ **do**

$s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$ # pairwise similarity

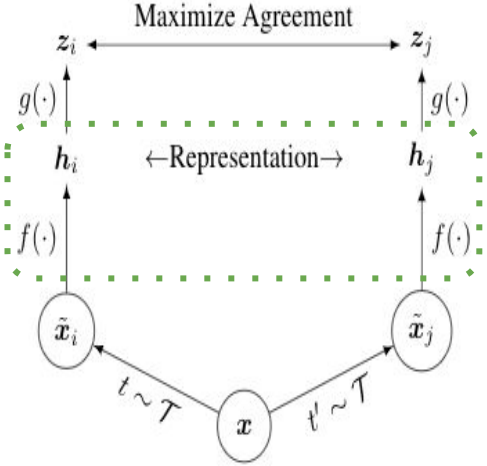
end for

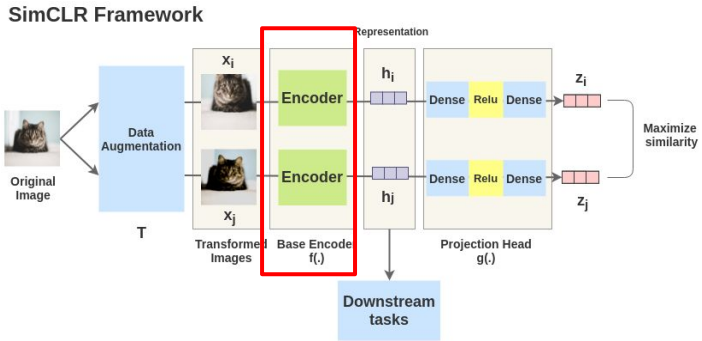
define $\ell(i,j)$ **as** $\ell(i,j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$

$\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$

update networks f and g to minimize $\mathcal{L}$

Results on leveraging the learned feature representations





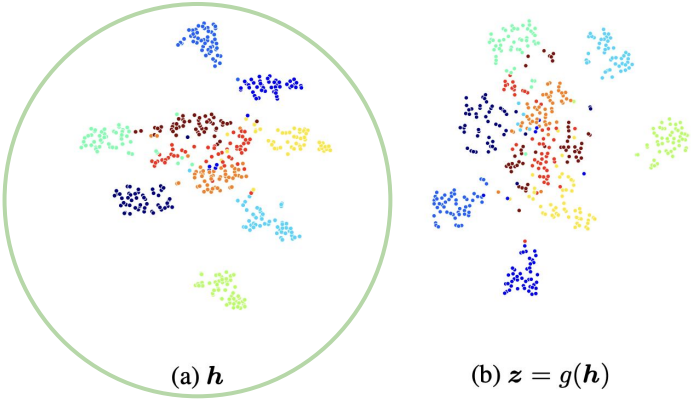


Figure B.4. t-SNE visualizations of hidden vectors of images from a randomly selected 10 classes in the validation set.

Results on leveraging the learned feature representations

Method	Architecture	Param (M)	Top 1	Top 5
Methods using ResNet-50:				
Local Agg.	ResNet-50	24	60.2	-
MoCo	ResNet-50	24	60.6	-
PIRL	ResNet-50	24	63.6	-
CPC v2	ResNet-50	24	63.8	85.3
SimCLR (ours)	ResNet-50	24	**69.3**	**89.0**
Methods using other architectures:				
Rotation	RevNet-50 (4×)	86	55.4	-
BigBiGAN	RevNet-50 (4×)	86	61.3	81.9
AMDIM	Custom-ResNet	626	68.1	-
CMC	ResNet-50 (2×)	188	68.4	88.2
MoCo	ResNet-50 (4×)	375	68.6	-
CPC v2	ResNet-161 (∗)	305	71.5	90.1
SimCLR (ours)	ResNet-50 (2×)	94	74.2	92.0
SimCLR (ours)	ResNet-50 (4×)	375	**76.5**	**93.2**

Table 6. ImageNet accuracies of linear classifiers trained on representations learned with different self-supervised methods.

Method	Architecture	Label fraction (Top 5) 1%	10%
Supervised baseline	ResNet-50	48.4	80.4
Methods using other label-propagation:			
Pseudo-label	ResNet-50	51.6	82.4
VAT+Entropy Min.	ResNet-50	47.0	83.4
UDA (w. RandAug)	ResNet-50	-	88.5
FixMatch (w. RandAug)	ResNet-50	-	89.1
S4L (Rot+VAT+En. M.)	ResNet-50 (4×)	-	91.2
Methods using representation learning only:			
InstDisc	ResNet-50	39.2	77.4
BigBiGAN	RevNet-50 (4×)	55.2	78.8
PIRL	ResNet-50	57.2	83.8
CPC v2	ResNet-161(∗)	77.9	91.2
SimCLR (ours)	ResNet-50	75.5	87.8
SimCLR (ours)	ResNet-50 (2×)	83.0	91.2
SimCLR (ours)	ResNet-50 (4×)	**85.8**	**92.6**

Table 7. ImageNet accuracy of models trained with few labels.

Key contributions/Impact of SimCLR

SimCLR has achieved state-of-the-art results on several benchmark datasets, demonstrating its effectiveness for self-supervised learning of visual representations.

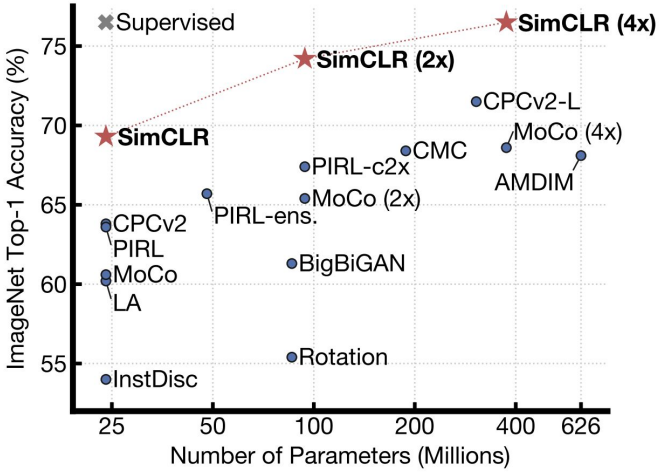


Figure 1. ImageNet Top-1 accuracy of linear classifiers trained on representations learned with different self-supervised methods (pretrained on ImageNet). Gray cross indicates supervised ResNet-50. Our method, SimCLR, is shown in bold.

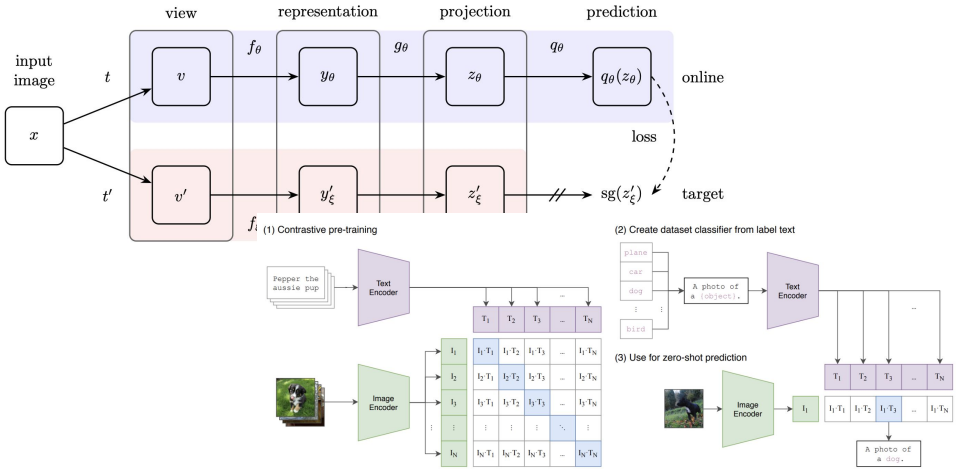
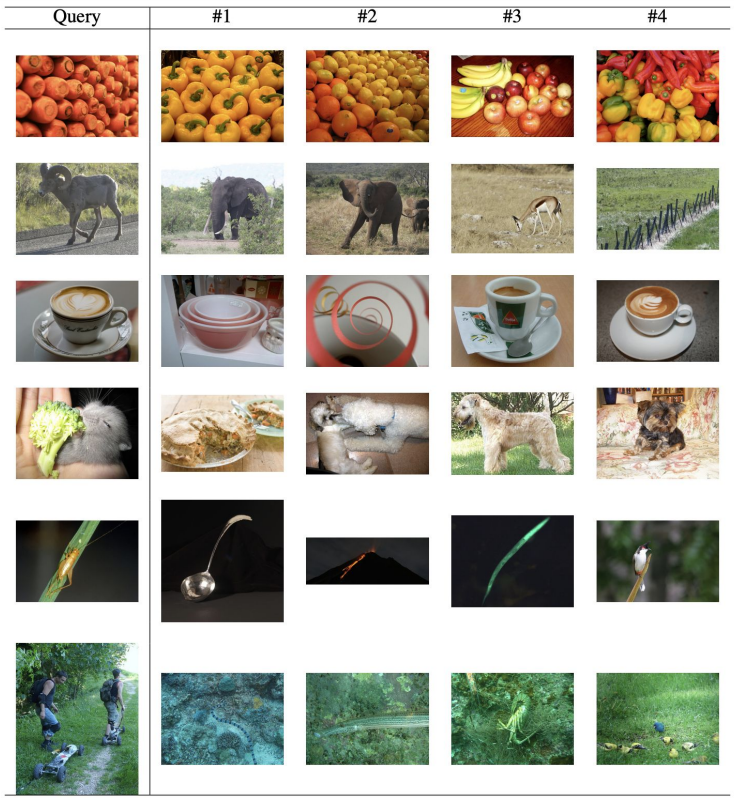


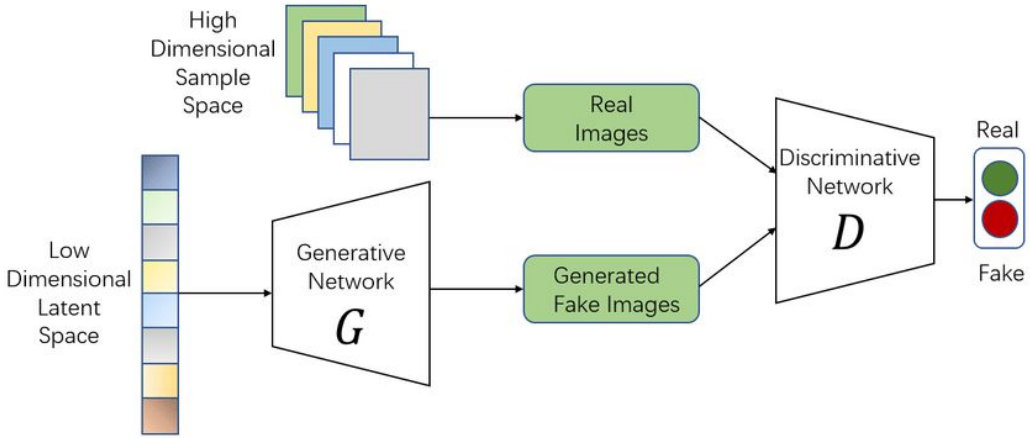
Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

Adversarial Feature Learning



What is Generative Adversarial Network (GAN)?

What is Generative Adversarial Network (GAN)?



$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Bidirectional Generative Adversarial Networks(BiGAN)

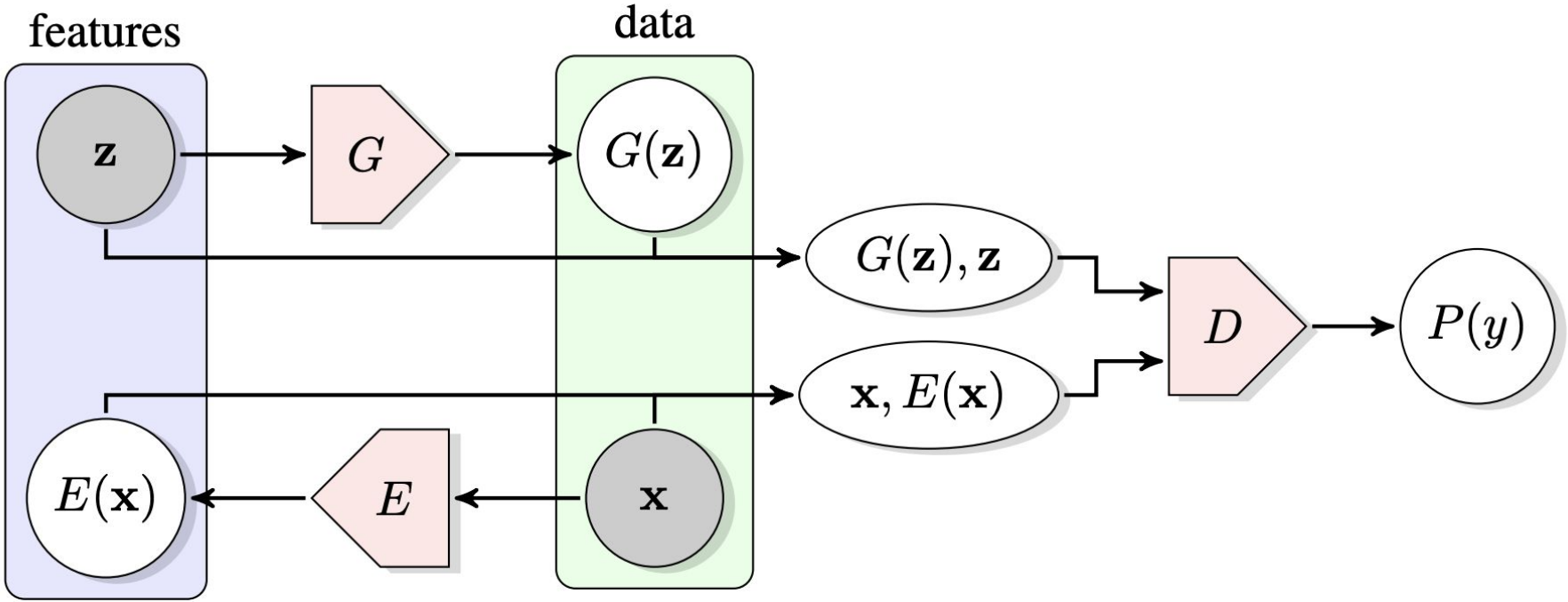


Figure 1: The structure of Bidirectional Generative Adversarial Networks (BiGAN).

Bidirectional Generative Adversarial Networks(BiGAN)

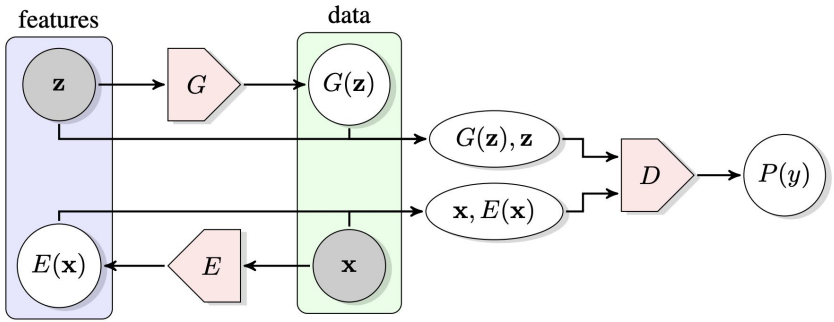


Figure 1: The structure of Bidirectional Generative Adversarial Networks (BiGAN).

The BiGAN training objective is defined as a minimax objective

$$\min_{G,E} \max_{D} V(D, E, G) \tag{2}$$

where

$$V(D,E,G) := \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{X}}}\Big[\underbrace{\mathbb{E}_{\mathbf{z}\sim p_E(\cdot|\mathbf{x})}\left[\log D(\mathbf{x},\mathbf{z})\right]}_{\log D(\mathbf{x},E(\mathbf{x}))}\Big] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{Z}}}\Big[\underbrace{\mathbb{E}_{\mathbf{x}\sim p_G(\cdot|\mathbf{z})}\left[\log\left(1-D(\mathbf{x},\mathbf{z})\right)\right]}_{\log(1-D(G(\mathbf{z}),\mathbf{z}))}\Big]. \tag{3}$$

Bidirectional Generative Adversarial Networks(BiGAN)

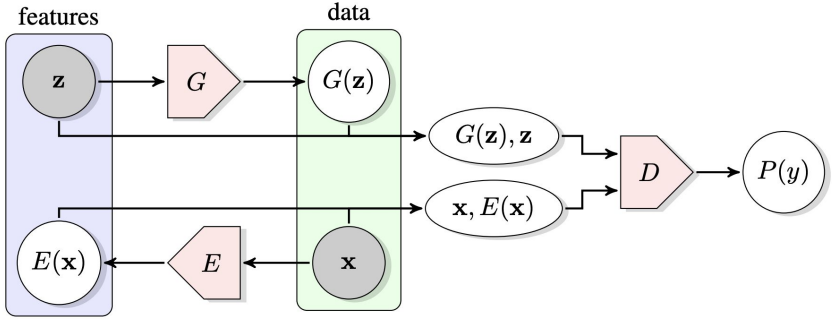


Figure 1: The structure of Bidirectional Generative Adversarial Networks (BiGAN).

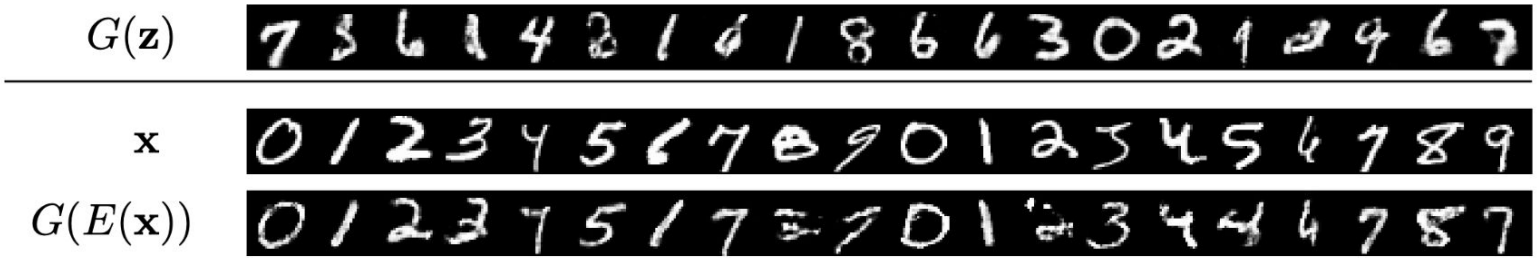


Figure 2: Qualitative results for permutation-invariant MNIST BiGAN training, including generator samples $G(\mathbf{z})$, real data $\mathbf{x}$, and corresponding reconstructions $G(E(\mathbf{x}))$.

Evaluation: Feature Representation

BiGAN	D	LR	JLR	AE (ℓ_2)	AE (ℓ_1)
97.39	97.30	97.44	97.13	97.58	97.63

Table 1: One Nearest Neighbors (1NN) classification accuracy (%) on the permutation-invariant MNIST (LeCun et al., 1998) test set in the feature space learned by BiGAN, Latent Regressor (LR), Joint Latent Regressor (JLR), and an autoencoder (AE) using an ℓ_1 or ℓ_2 distance.

Evaluation: Feature Representation

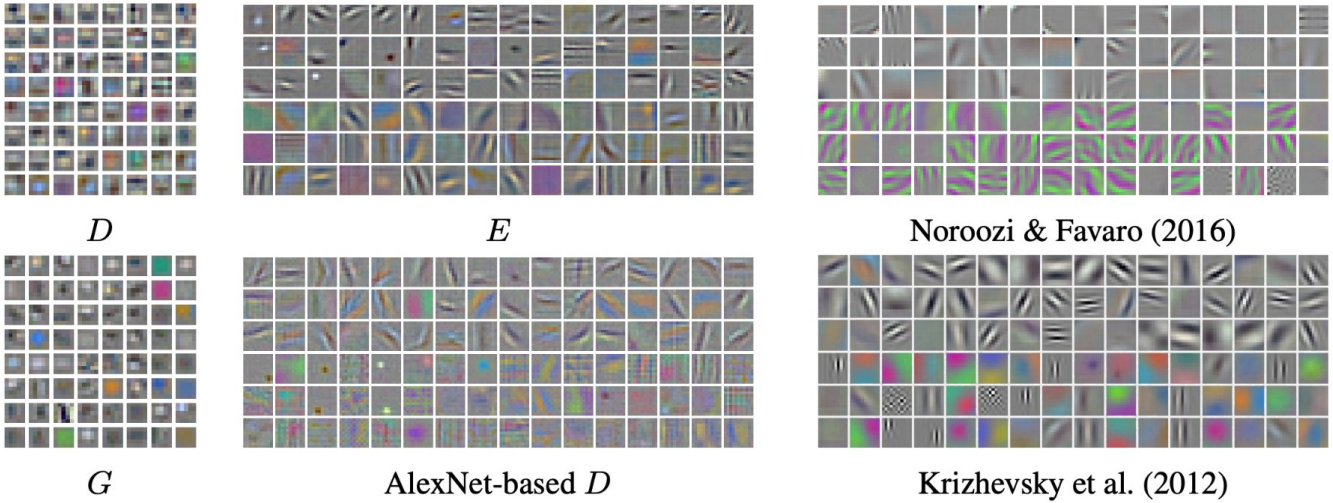


Figure 3: The convolutional filters learned by the three modules (D, G, and E) of a BiGAN (left, top-middle) trained on the ImageNet (Russakovsky et al., 2015) database. We compare with the filters learned by a discriminator D trained with the same architecture (bottom-middle), as well as the filters reported by Noroozi & Favaro (2016), and by Krizhevsky et al. (2012) for fully supervised ImageNet training (right).

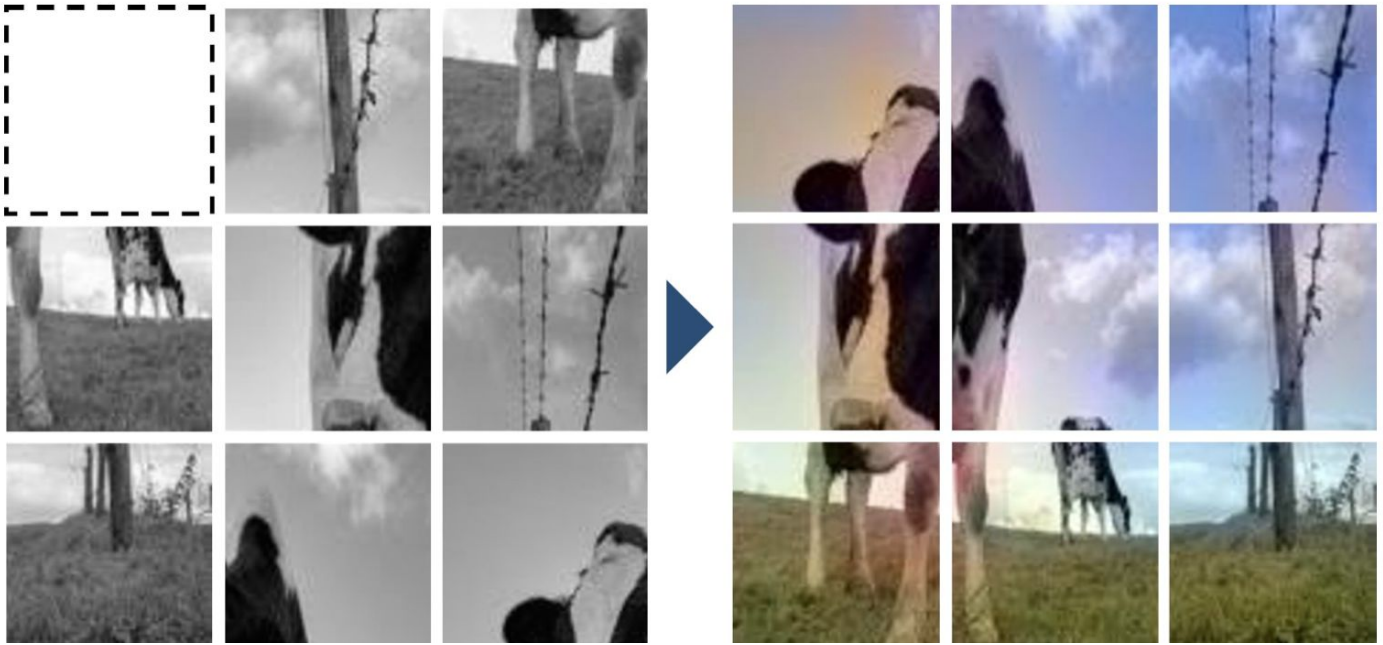
Evaluation: Feature Representation

		Classification (% mAP)			*FRCN* Detection (% mAP)	*FCN* Segmentation (% mIU)
	trained layers	fc8	fc6-8	all	all	all
sup.	ImageNet (Krizhevsky et al., 2012)	**77.0**	**78.8**	**78.3**	**56.8**	**48.0**
self-sup.	Agrawal et al. (2015)	31.2	31.0	54.2	43.9	-
	Pathak et al. (2016)	30.5	34.6	56.5	44.5	**30.0**
	Wang & Gupta (2015)	28.4	**55.6**	63.1	47.4	-
	Doersch et al. (2015)	**44.7**	55.1	**65.3**	**51.1**	-
unsup.	k-means (Krähenbühl et al., 2016)	32.0	39.2	56.6	45.6	32.6
	Discriminator (D)	30.7	40.5	56.4	-	-
	Latent Regressor (LR)	36.9	47.9	57.1	-	-
	Joint LR	37.1	47.9	56.5	-	-
	Autoencoder (ℓ_2)	24.8	16.0	53.8	41.9	-
	BiGAN (ours)	37.5	48.7	58.9	46.2	34.9
	BiGAN, 112×112 E (ours)	**41.7**	**52.5**	**60.3**	**46.9**	**35.2**

Paper 4.3

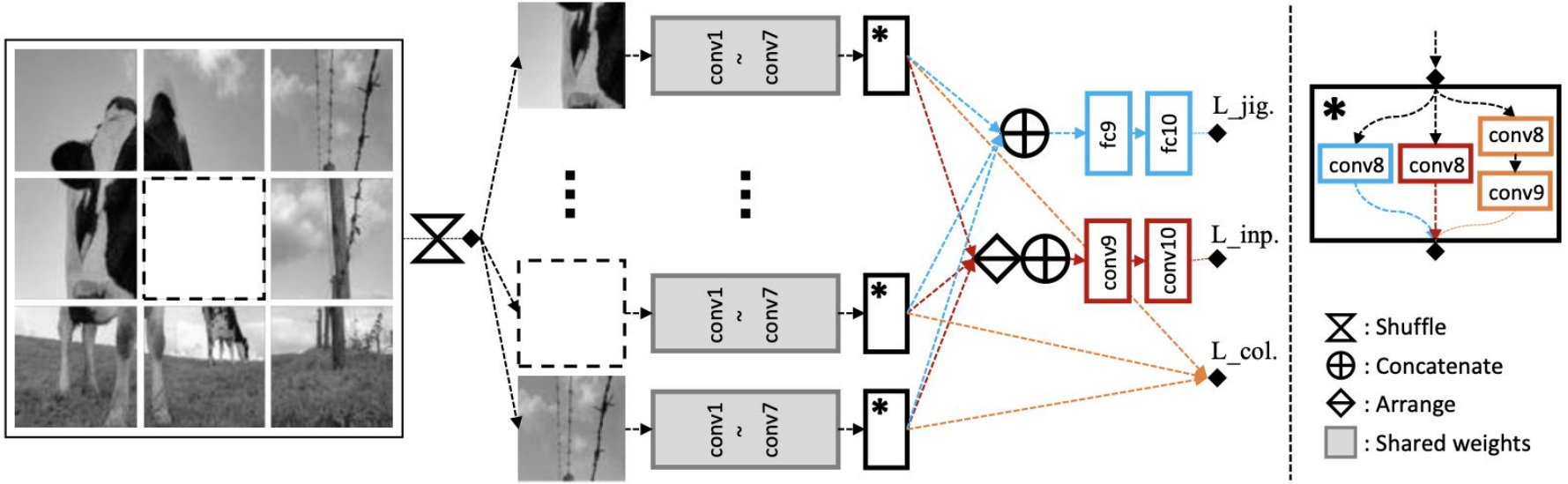
Placeholder for Yousif Alozairi

Learning Image Representations by Completing Damaged Jigsaw Puzzles (2018)



What does it mean to Complete Damaged Jigsaw Puzzles?

- Rearranging
- Inpainting
- Colorization



How Did They Complicate Self-Supervised tasks?

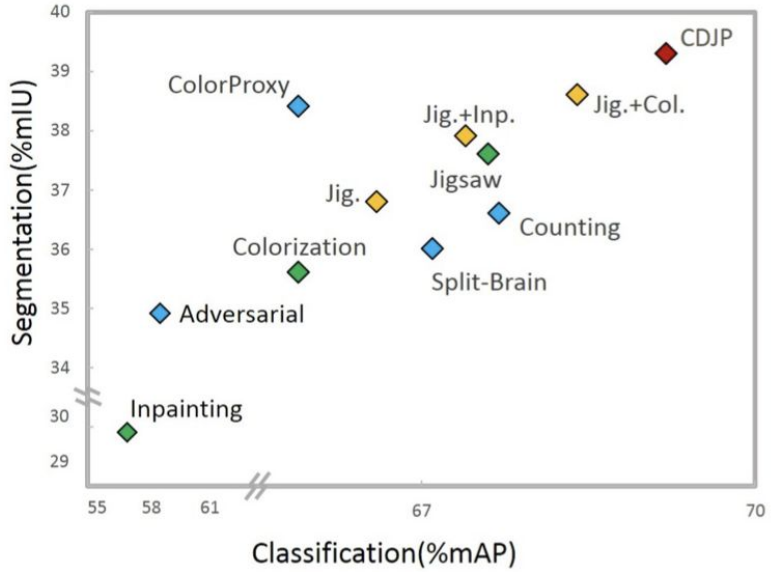
The Tasks of CDJP:

- Rearranging: One piece is removed, the rest are rearranged
- Inpainting: The missing patch must be generated from surrounding context
- Colorization: The network sees a decolorized image and must colorize it

- With the combination of these three, CDJP outperforms previous self-supervised learning methods in image classification, object detection, and segmentation (PASCAL VOC, ImageNet)

Damage-and-recover Self-Supervised Learning

- Each form of damage teaches different lessons/levels of understanding
- Three types of damage in a single task complicate the process
- This approach maximized effectiveness as shown by CDJP:



The Effect on Classification & Segmentation

Combination	Class.	Segm.
Jig.	66.6	36.8
Jig.+Inp.	67.4	37.9
Jig.+Col.	68.4	38.6
Jig.+Inp.+Col./simple	68.0	38.1
Jig.+Inp.+Col.(CDJP)	**69.2**	**39.3**

- The final method combining all three complicated tasks obtains the best scores when compared with PASCAL
- It improves over jigsaw capabilities alone by 2.6% & 2.5% both in classification and semantic segmentation task

Nearest Neighbor Search

Image retrieval results compared:

semantic understanding of shapes/poses despite different colors

jigsaw puzzle, inpainting, colorization, CDJP, and ImageNet classification respectively

Lessons Learned

- CDJP (Completing damaged jigsaw puzzles) showcases how more complicated, multi-step problems for self-supervised representation learning can achieve more generalized and often more effective results:
    - Combining tasks consistently increased performance of their model over simpler tasks/combinations
- Hence, harder tasks can lead to better representations, so damaging image data then training neural networks to recover it is a successful improvement

A critical analysis of self-supervision

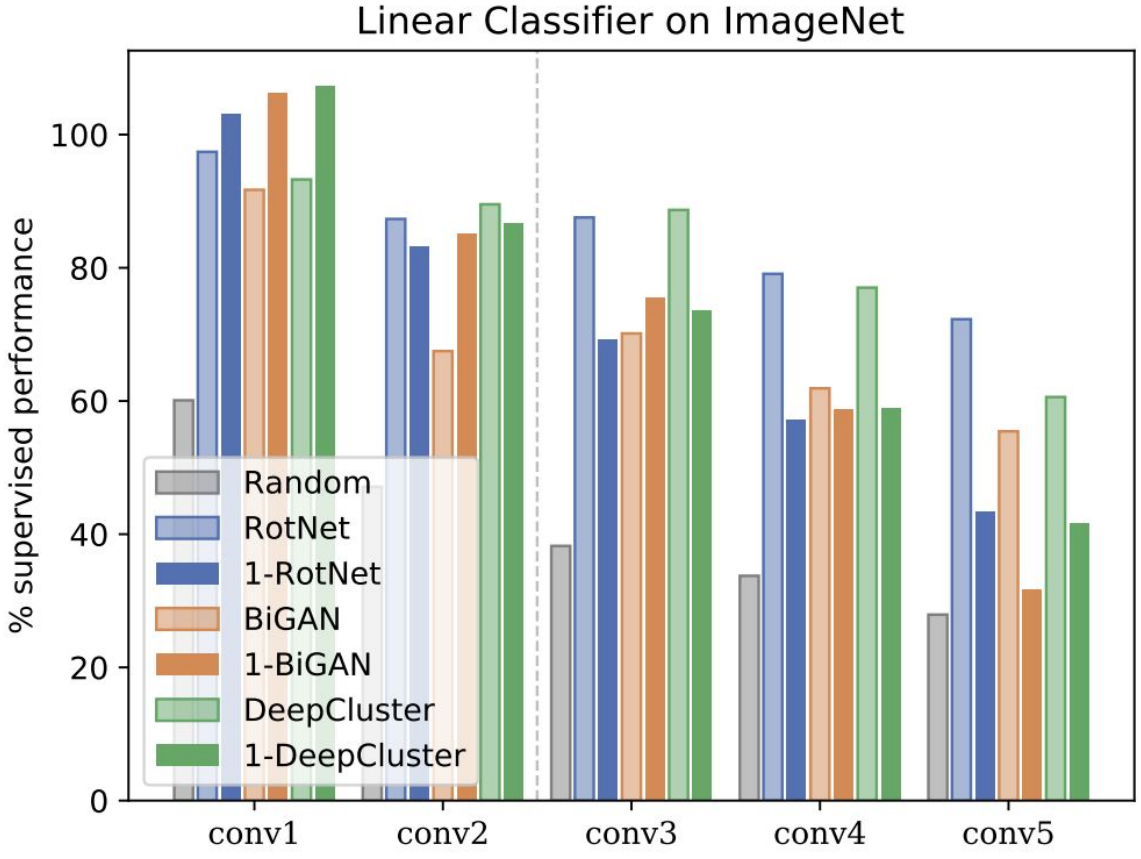
(University of Oxford)

Jian Zhang


A
自転車専用
アメ横
マルセル
御徒町店


B


C


Linear Classifier on ImageNet
% supervised performance
0
20
40
60
80
100
conv1
conv2
conv3
conv4
conv5
Random
RotNet
1-RotNet
BiGAN
1-BiGAN
DeepCluster
1-DeepCluster

REPRESENTATION LEARNING METHODS

- Generative Adversarial Networks (GANs)
- Bidirectional Generative Adversarial Networks (BiGAN)
- RotNet
- DeepCluster

	Method, Reference	#images	ILSVRC-12 conv1	conv2	conv3	conv4	conv5
(a)	Full-supervision‡	1,281,167	19.3	36.3	44.2	48.3	50.5
(b)	(Oyallon et al., 2017): Scattering	0	-	18.9	-	-	-
(c)	Random‡	0	11.6	17.1	16.9	16.3	14.1
(d)	(Krähenbühl et al., 2016):k-means‡	$\approx$160	17.5	23.0	24.5	23.2	20.6
(e)	(Donahue et al., 2017): BiGAN‡	1,281,167	17.7	24.5	31.0	29.9	28.0
(f)	mono, Image A	1	20.4	30.9	33.4	28.4	16.0
(g)	mono, Image B	1	20.5	30.4	31.6	27.0	16.8
(h)	deka	10	16.2	16.5	16.5	13.1	7.5
(i)	kilo	1,000	16.1	17.7	18.3	17.6	13.5
(j)	(Gidaris et al., 2018): RotNet	1,281,167	18.8	31.7	38.7	38.2	36.5
(k)	mono, Image A	1	19.9	30.2	30.6	27.6	21.9
(l)	mono, Image B	1	17.8	27.6	27.9	25.4	20.2
(m)	deka	10	19.6	30.7	32.6	28.9	22.6
(n)	kilo	1,000	21.0	33.5	36.5	34.0	29.4
(o)	(Caron et al., 2018): DeepCluster	1,281,167	18.0	32.5	39.2	37.2	30.6
(p)	mono, Image A	1	20.7	31.5	32.5	28.5	21.0
(q)	mono, Image B	1	19.7	30.1	31.6	28.5	20.4
(r)	mono, Image C	1	18.9	29.2	31.5	28.9	23.5
(s)	deka	10	18.5	29.0	31.1	28.2	21.9
(t)	kilo	1,000	19.5	29.8	33.0	31.7	26.8

Table 4: **Finetuning experiments** The pretrained model's first two convolutions are left frozen (or replaced by the Scattering transform) and the nework is retrained using ImageNet LSVRC-12 training set.

	Top-1
Full sup.	59.4
Random	42.6
Scattering	49.2
BiGAN, A	51.4
RotNet, A	49.5
DeepCluster A	52.5

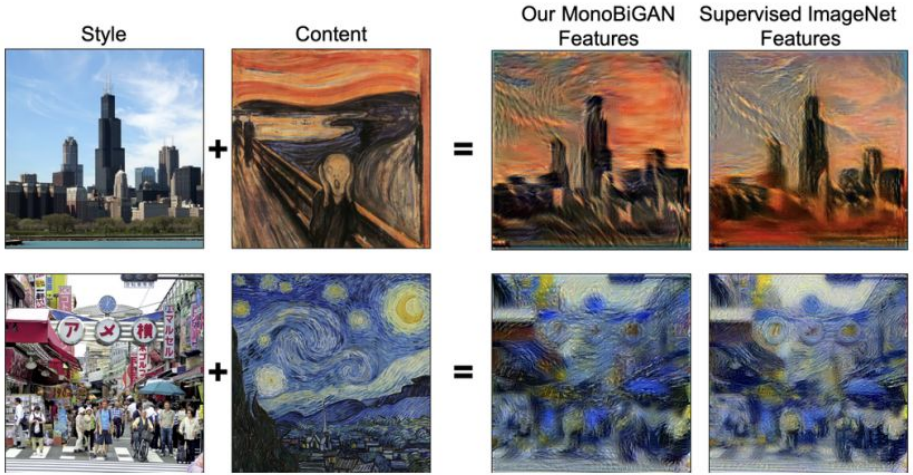
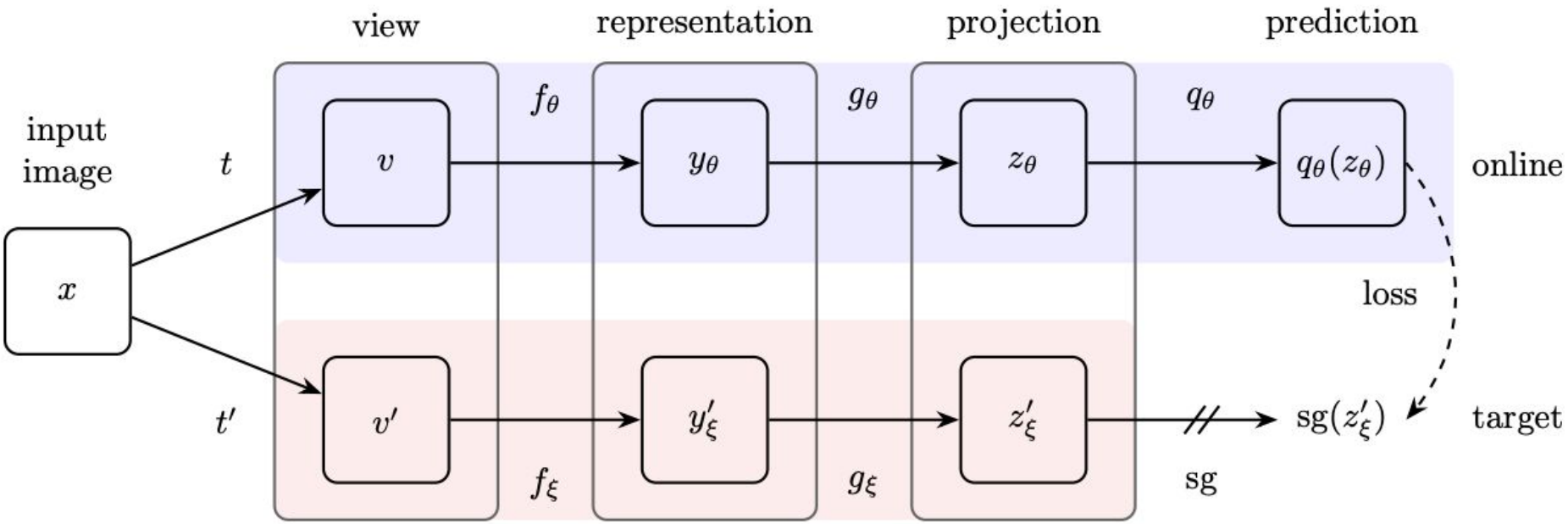


Figure 3: **Style transfer with single-image pretraining.** We show two style transfer results using the Image A trained BiGAN and the ImageNet pretrained AlexNet.

Bootstrap Your Own Latent (BYOL, DeepMind)

Jian Zhang

BYOL is **NOT** a contrastive learning method



Loss:

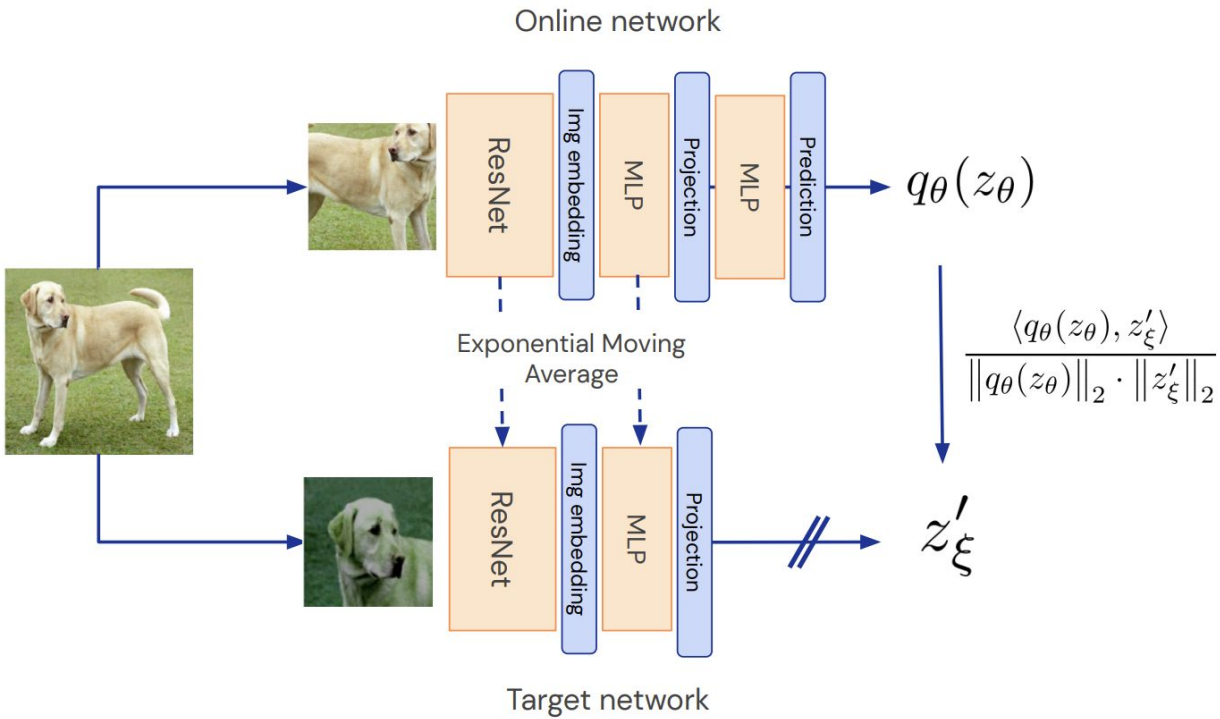
$$\mathcal{L}_{\theta,\xi} \triangleq \left\| \overline{q_\theta}(z_\theta) - \overline{z}'_\xi \right\|_2^2 = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\left\| q_\theta(z_\theta) \right\|_2 \cdot \left\| z'_\xi \right\|_2}$$

$$\mathcal{L}^{\mathtt{BYOL}}_{\theta,\xi} = \mathcal{L}_{\theta,\xi} + \widetilde{\mathcal{L}}_{\theta,\xi}$$

Train:

$$\theta \leftarrow \mathrm{optimizer}\left(\theta, \nabla_\theta \mathcal{L}^{\mathtt{BYOL}}_{\theta,\xi}, \eta\right)$$
$$\xi \leftarrow \tau\xi + (1-\tau)\theta,$$

Intuition


Online network
ResNet
Img embedding
MLP
Projection
MLP
Prediction
$q_\theta(z_\theta)$
Exponential Moving Average
$\frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\|q_\theta(z_\theta)\|_2 \cdot \|z'_\xi\|_2}$
ResNet
Img embedding
MLP
Projection
z'_ξ
Target network

Result

Method	Top-1	Top-5
Local Agg.	60.2	-
PIRL [35]	63.6	-
CPC v2 [32]	63.8	85.3
CMC [11]	66.2	87.0
SimCLR [8]	69.3	89.0
MoCo v2 [37]	71.1	-
InfoMin Aug. [12]	73.0	91.1
BYOL (ours)	**74.3**	**91.6**

(a) ResNet-50 encoder.

Method	Architecture	Param.	Top-1	Top-5
SimCLR [8]	ResNet-50 (2×)	94M	74.2	92.0
CMC [11]	ResNet-50 (2×)	94M	70.6	89.7
BYOL (ours)	ResNet-50 (2×)	94M	77.4	93.6
CPC v2 [32]	ResNet-161	305M	71.5	90.1
MoCo [9]	ResNet-50 (4×)	375M	68.6	-
SimCLR [8]	ResNet-50 (4×)	375M	76.5	93.2
BYOL (ours)	ResNet-50 (4×)	375M	78.6	94.2
BYOL (ours)	ResNet-200 (2×)	250M	**79.6**	**94.8**

(b) Other ResNet encoder architectures.

Table 1: Top-1 and top-5 accuracies (in %) under linear evaluation on ImageNet.

Result

Semi-supervised training on ImageNet

Method	Top-1		Top-5	
	1%	10%	1%	10%
Supervised [77]	25.4	56.4	48.4	80.4
InstDisc	-	-	39.2	77.4
PIRL [35]	-	-	57.2	83.8
SimCLR [8]	48.3	65.6	75.5	87.8
BYOL (ours)	**53.2**	**68.8**	**78.4**	**89.0**

(a) ResNet-50 encoder.

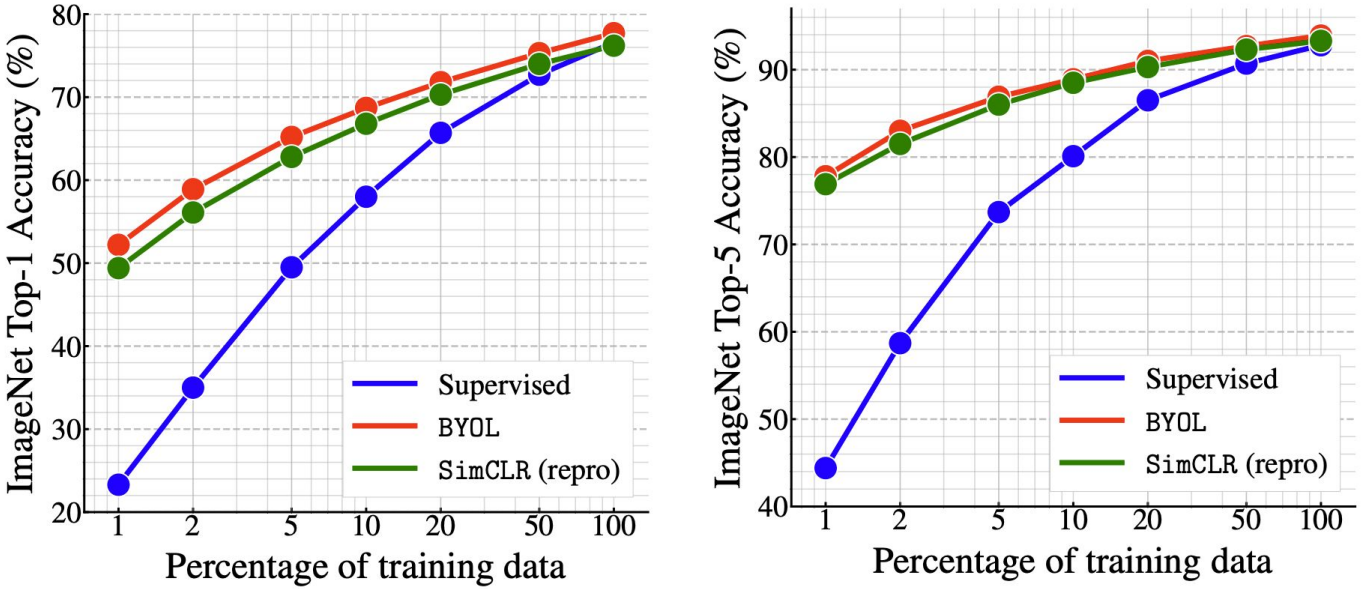
Method	Architecture	Param.	Top-1		Top-5	
			1%	10%	1%	10%
CPC v2 [32]	ResNet-161	305M	-	-	77.9	91.2
SimCLR [8]	ResNet-50 (2×)	94M	58.5	71.7	83.0	91.2
BYOL (ours)	ResNet-50 (2×)	94M	62.2	73.5	84.1	91.7
SimCLR [8]	ResNet-50 (4×)	375M	63.0	74.4	85.8	92.6
BYOL (ours)	ResNet-50 (4×)	375M	69.1	75.7	87.9	92.5
BYOL (ours)	ResNet-200 (2×)	250M	**71.2**	**77.7**	**89.5**	**93.7**

(b) Other ResNet encoder architectures.

Table 2: Semi-supervised training with a fraction of ImageNet labels.

Result

Semi-supervised training on ImageNet



(a) Top-1 accuracy

(b) Top-5 accuracy

Figure 4: Semi-supervised training with a fraction of ImageNet labels on a ResNet-50 ($\times 1$).

Conclusion

+ BYOL learns its representation by predicting previous versions of its outputs, without using negative pairs.

+ BYOL bridges most of the remaining gap between self-supervised methods and the supervised learning baseline

- Sensitive to batch size & optimizer

BYOL works even without batch statistics

(https://arxiv.org/pdf/2010.10241)

Paper 4.7 & 4.8

Placeholder for Niranjan Sundararajan

Unsupervised Learning of Video Representations using LSTMs

Niranjan Sundararajan, https://arxiv.org/pdf/1502.04681

Motivation

Video data is inherently sequential; understanding temporal dependencies is crucial.

Supervised learning for video representation is expensive and requires extensive labeled data.

Goal: Learn meaningful video representations in an unsupervised manner using LSTMs.

Methodology

Encoder-Decoder LSTM Architecture

- Encoder LSTM: Extracts a fixed-length representation from input sequences.
- Decoder LSTM(s): Reconstructs input or predicts future frames.

Trained on two types of inputs:

- Raw Image Patches (pixel-level data)
- High-Level Features (extracted via CNNs)

LSTM Architecture

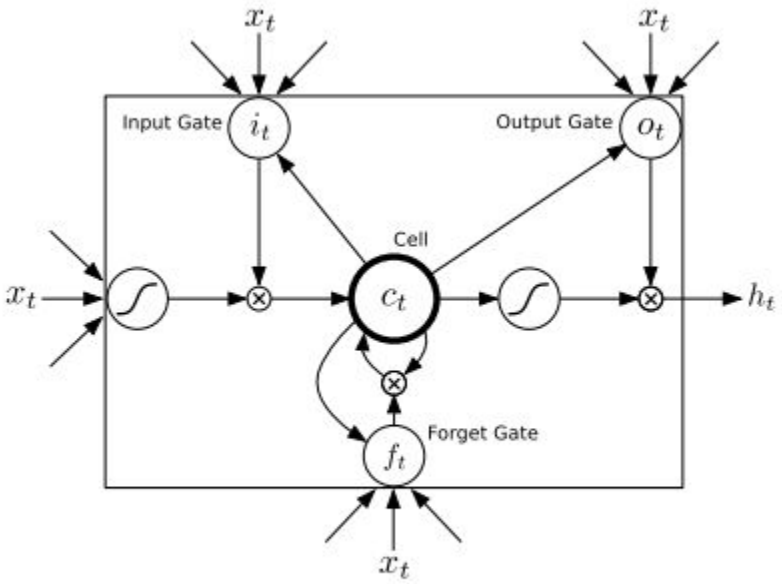


Figure 1. LSTM unit

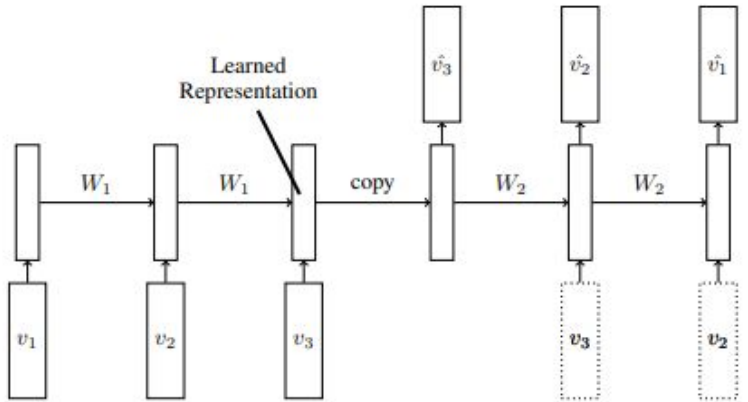


Figure 2. LSTM Autoencoder Model

LSTM Architecture (contd.)

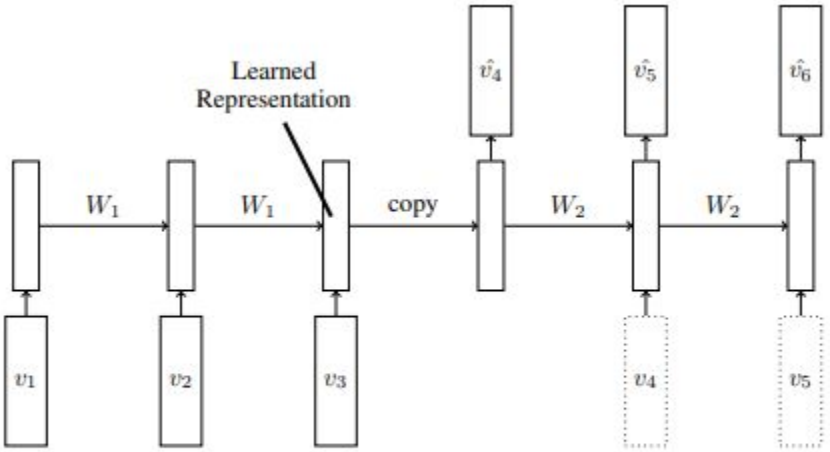


Figure 3. LSTM Future Predictor Model

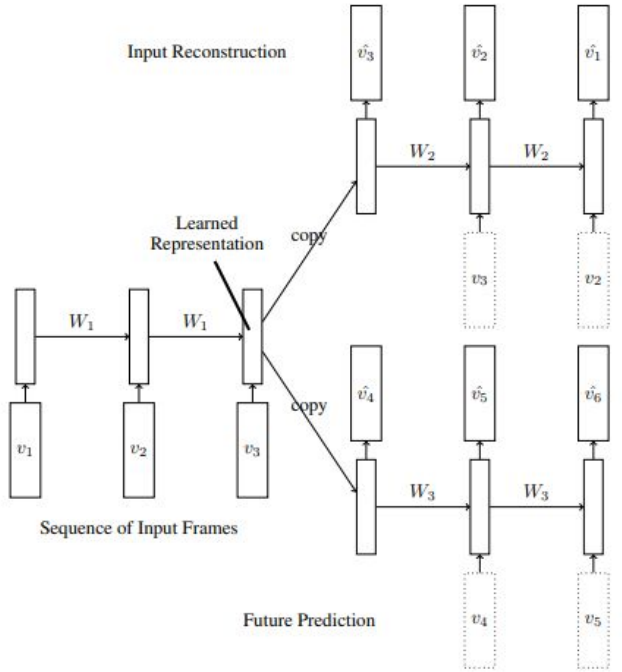


Figure 4. The Composite Model: The LSTM predicts the future as well as the input sequence.

LSTM Architecture (contd.)

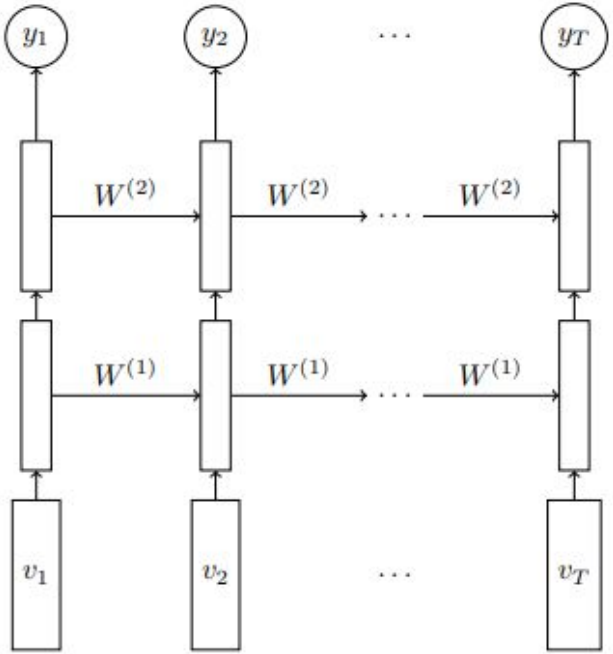
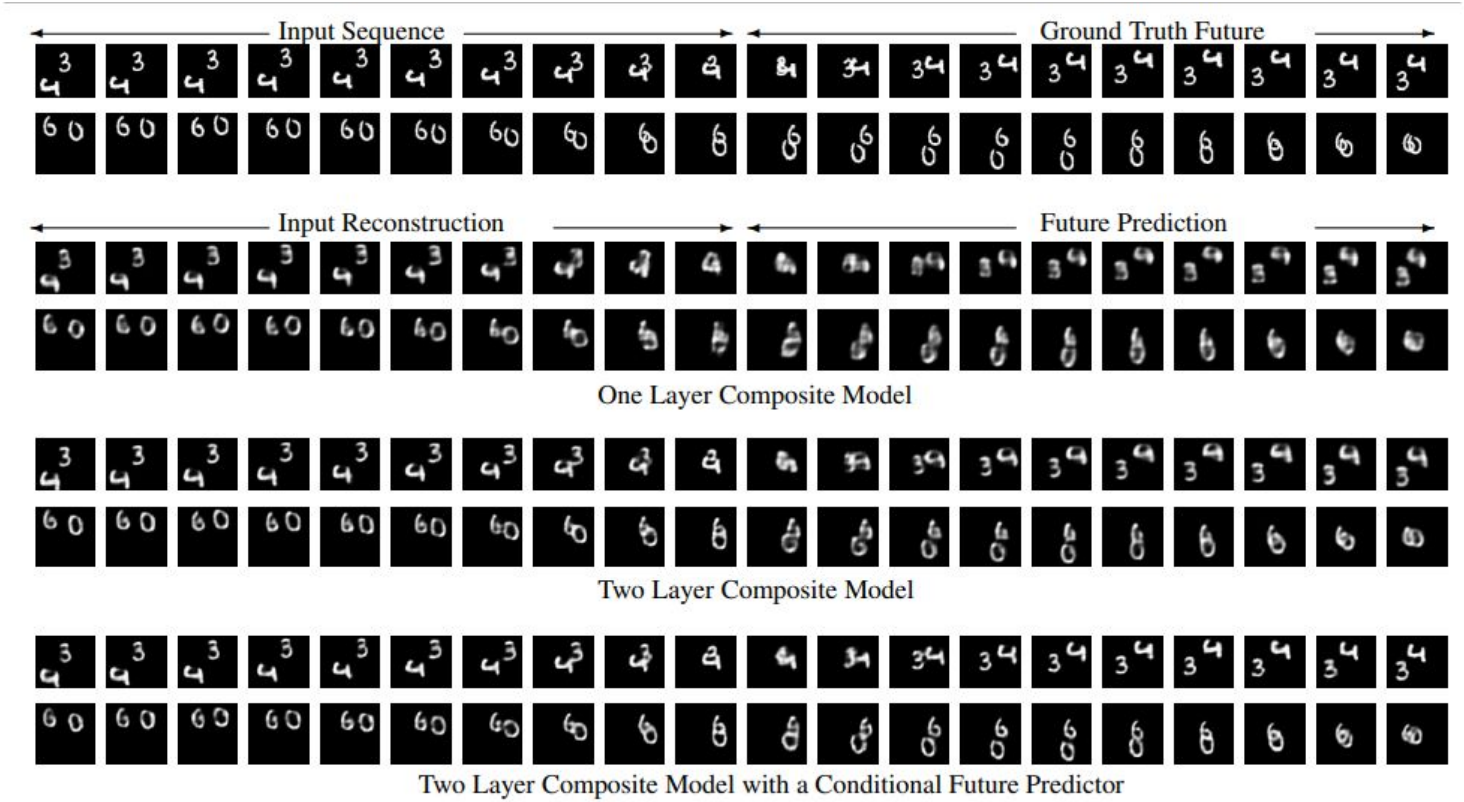
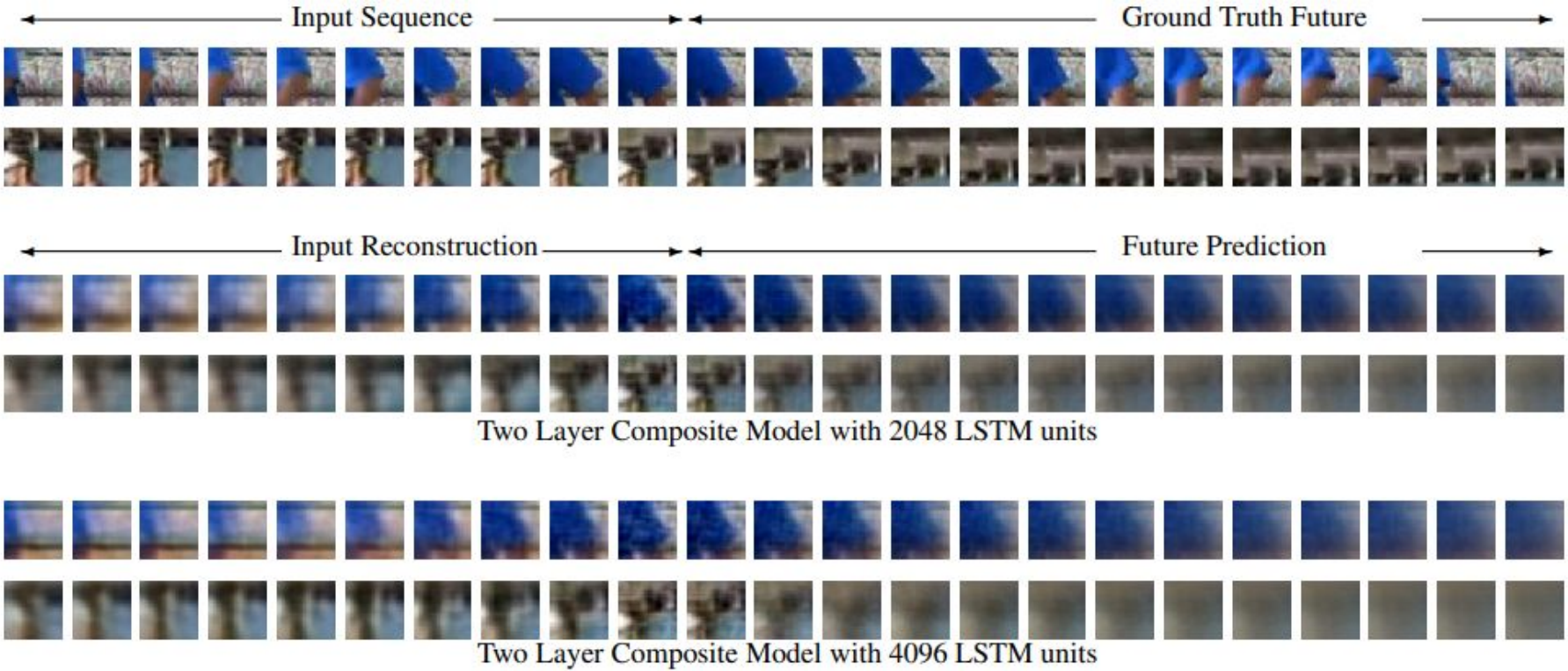


Figure 11. LSTM Classifier.

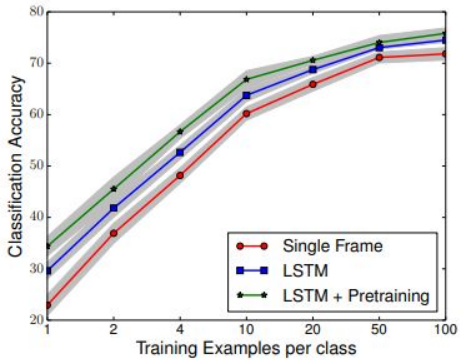
Results

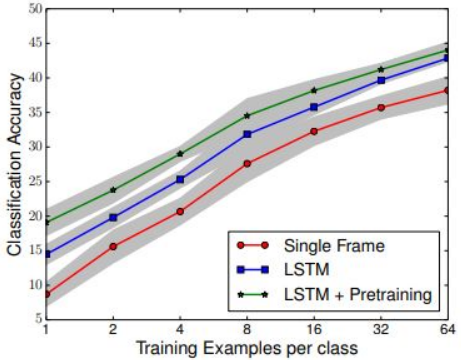

Input Sequence
Ground Truth Future
Input Reconstruction
Future Prediction
One Layer Composite Model
Two Layer Composite Model
Two Layer Composite Model with a Conditional Future Predictor

Results (contd.)



Results (contd.)





Method	UCF-101 small	UCF-101	HMDB-51 small	HMDB-51
Baseline LSTM	63.7	74.5	25.3	42.8
Autoencoder	66.2	75.1	28.6	44.0
Future Predictor	64.9	74.9	27.3	43.1
Conditional Autoencoder	65.8	74.8	27.9	43.1
Conditional Future Predictor	65.1	74.9	27.4	43.4
Composite Model	67.0	**75.8**	29.1	**44.1**
Composite Model with Conditional Future Predictor	**67.1**	**75.8**	**29.2**	44.0

Results (contd.)

Method	UCF-101	HMDB-51
Spatial Convolutional Net (Simonyan & Zisserman, 2014a)	73.0	40.5
C3D (Tran et al., 2014)	72.3	-
C3D + fc6 (Tran et al., 2014)	**76.4**	-
LRCN (Donahue et al., 2014)	71.1	-
Composite LSTM Model	75.8	44.0
Temporal Convolutional Net (Simonyan & Zisserman, 2014a)	**83.7**	54.6
LRCN (Donahue et al., 2014)	77.0	-
Composite LSTM Model	77.7	-
LRCN (Donahue et al., 2014)	82.9	-
Two-stream Convolutional Net (Simonyan & Zisserman, 2014a)	88.0	59.4
Multi-skip feature stacking (Lan et al., 2014)	**89.1**	**65.1**
Composite LSTM Model	84.3	-

Tracking Emerges by Colorizing Videos

Niranjan Sundararajan, https://arxiv.org/pdf/1806.09594

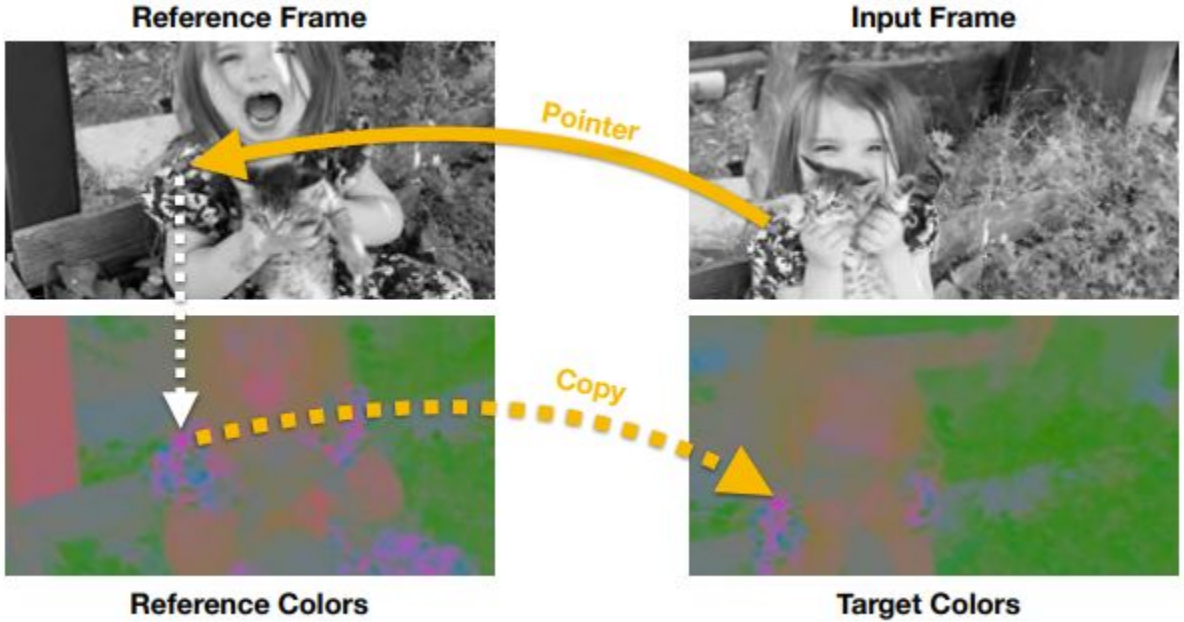
Motivation

Traditional tracking requires labeled datasets (expensive & time-consuming)

Can we learn to track without annotations?

Idea: If a model can colorize video frames using a reference frame, it must have learned to track objects.

Methodology



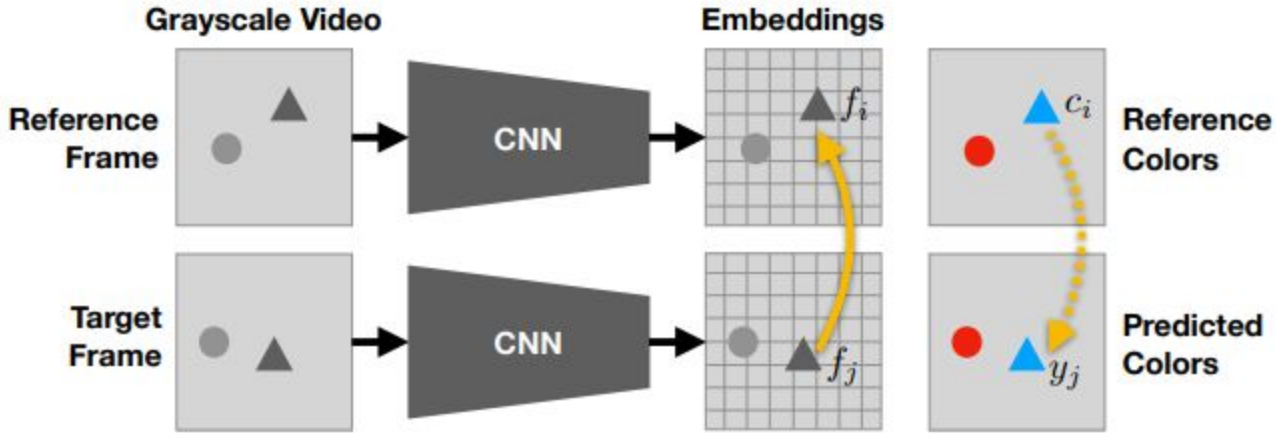
Methodology

Self-Supervised Task: Colorizing Grayscale Video Frames

- Given a grayscale frame and a colored reference frame, the model must transfer colors.
- To succeed, the model must track objects across frames.

Tracking emerges as a byproduct of colorization.

Methodology (contd.)

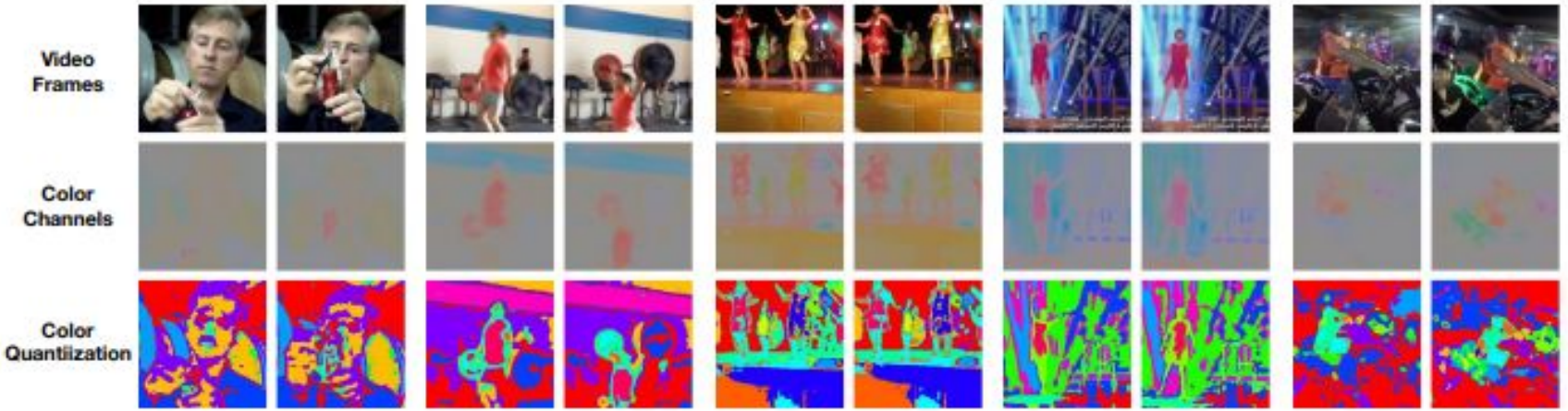


Methodology (contd.)

$$A_{ij} = \frac{\exp\left(f_i^T f_j\right)}{\sum_k \exp\left(f_k^T f_j\right)}$$

$$y_j = \sum_i A_{ij} c_i$$

$$\min_\theta \sum_j \mathcal{L}\left(y_j, c_j\right)$$



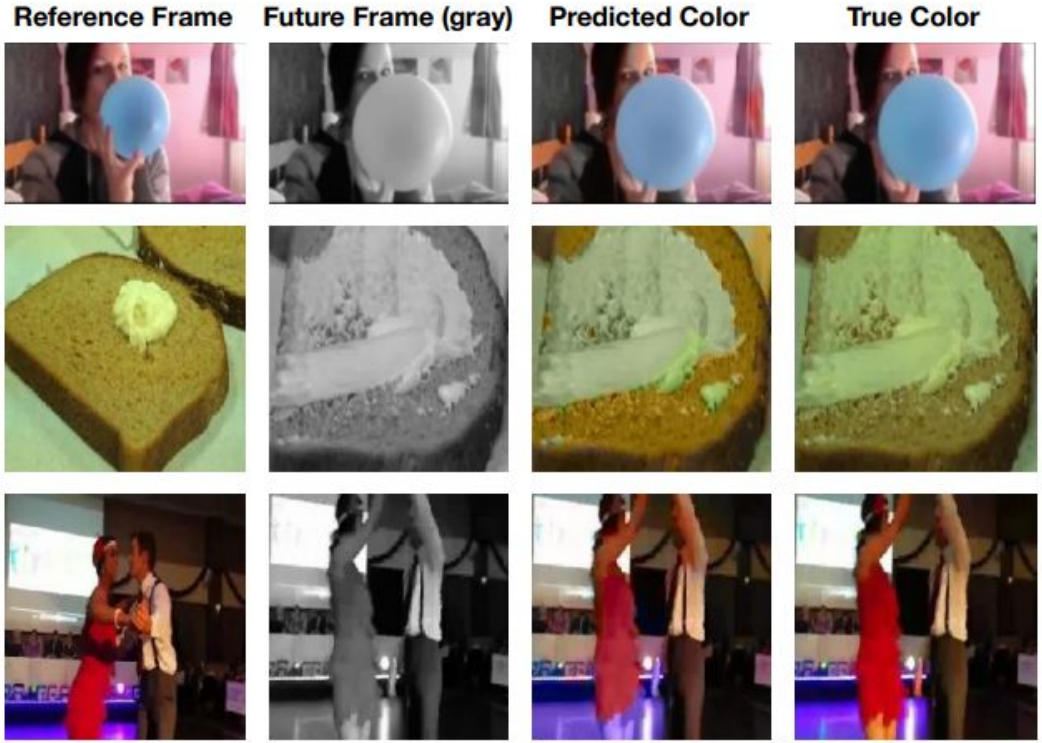
Why does Tracking Emerge?

Objects move smoothly across frames → Model learns motion consistency.

Correct colorization requires accurate pixel correspondence (i.e., tracking).

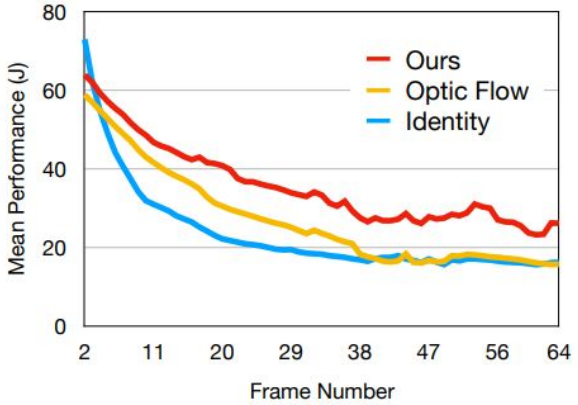
Learned representations can be directly used for tracking tasks.

Results

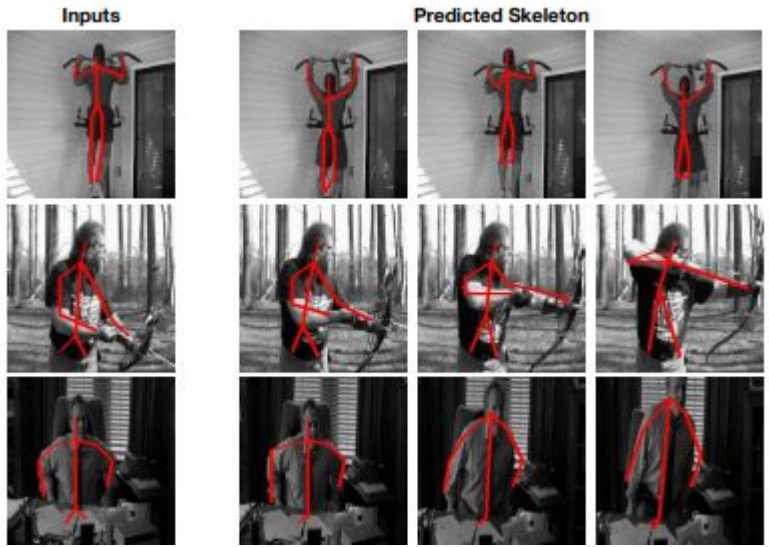

Reference Frame
Future Frame (gray)
Predicted Color
True Color

Results (contd.)

Method	Supervised?	Segment	Boundary
Identity		22.1	23.6
Single Image Colorization		4.7	5.2
Optical Flow (Coarse-to-Fine) [59]		13.0	15.1
Optical Flow (FlowNet2) [23]		26.7	25.2
Ours		34.6	32.7
Fully Supervised [46, 47]	✓	55.1	62.1



Results (contd.)



Paper 4.10

Placeholder for Yousif Alozairi

Video Representation Learning by Dense Predictive Coding (2019)

(a)

(b)

What is Dense Predictive Coding (DPC)?

- DPC is ultimately predicting future feature representations instead of raw frames

The Benefit:

- Avoids low-level pixel prediction, and complicated video labeling
- Instead, this encourages learning semantic representations

The Method:

- It encodes past video blocks
- It predicts future feature embeddings
- It uses contrastive loss to distinguish correct future from distractors

The Dense Predictive Coding Method

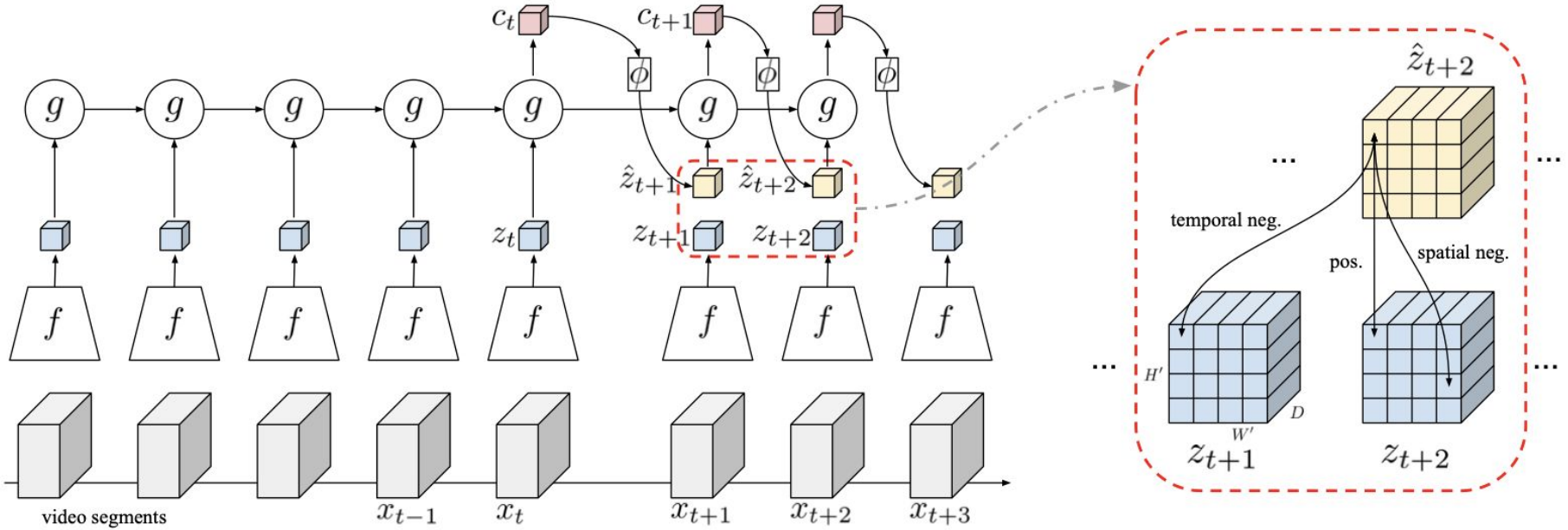


Figure 2: A diagram of **Dense Predictive Coding** method. The left part is the pipeline of the DPC, which is explained in Sec. 3.1. The right part (in the dashed rectangle) is an illustration of the Pred-GT pair construction for contrastive loss, which is explained in Sec. 3.2.

Progressive Learning

Future prediction is challenging:

- More uncertainty further into the future
- Low-level cues (optical flow) become unreliable

Hence, Progressive Learning:

- Start with short-term predictions
- Gradually increase prediction horizon (up to 2 seconds)
- Forces network to learn high-level semantics

Surpasses all previous self-supervised & pretrained methods

Self-Supervised Method (RGB stream only)			Supervised Accuracy (top1 acc)	
Method	Architecture (#param)	Dataset	UCF101	HMDB51
Random Initialization	3D-ResNet18 (14.2M)	-	46.5	17.1
ImageNet Pretrained [33]	VGG-M-2048 (25.4M)	-	73.0	40.5
Shuffle & Learn [27] (227 × 227)	CaffeNet (58.3M)	UCF101/HMDB51	50.2	18.1
OPN [22] (80 × 80)	VGG-M-2048 (8.6M)	UCF101/HMDB51	59.8	23.8
OPN [22] (120 × 120)	VGG-M-2048 (11.2M)	UCF101/HMDB51	55.4	-
OPN [22] (224 × 224)	VGG-M-2048 (25.4M)	UCF101/HMDB51	51.9	-
Ours (128 × 128)	3D-ResNet18 (14.2M)	UCF101	**60.6**	-
3D-RotNet [15] (112 × 112)	3D-ResNet18-full (33.6M)	Kinetics-400	62.9	33.7
3D-ST-Puzzle [17] (224 × 224)	3D-ResNet18-full (33.6M)	Kinetics-400	63.9 (65.8*)	33.7*
Ours (128 × 128)	3D-ResNet18 (14.2M)	Kinetics-400	**68.2**	**34.5**
Ours (224 × 224)	3D-ResNet34 (32.6M)	Kinetics-400	**75.7**	**35.7**

Table 4: Comparison with other self-supervised methods, results are reported as an average over three training-testing splits. Note that, previous works [15, 17] use full-scale 3D-ResNet18, *i.e.* all convolutions are 3D, and the input sizes for different models have been shown. *indicates the results from the multi-task self-supervised learning, *i.e.* Rotation + 3D Puzzle.

Nearest Neighbor Retrieval shows Semantic Understanding

(a)

(b)

Figure 4: More examples of video retrieval with nearest neighbour (same setting as Figure 1). Figure 4a is the NN retrieval with DPC pre-trained $f(.)$ on UCF101 (performance reported in Sec. 4.1.2). Figure 4b is the NN retrieval with ImageNet inflated $f(.)$. Retrieval is performed on UCF101 validation set.

Key Takeaways

- Dense Predictive Coding outperformed all self-supervised methods on video representation learning
- Predicting future embeddings seems to allow for overall better action understanding
- Future models should expand upon this for better qualitative results but also semantic understanding