Video Representation Learning by Dense Predictive Coding

Tengda Han, Weidi Xie, Andrew Zisserman

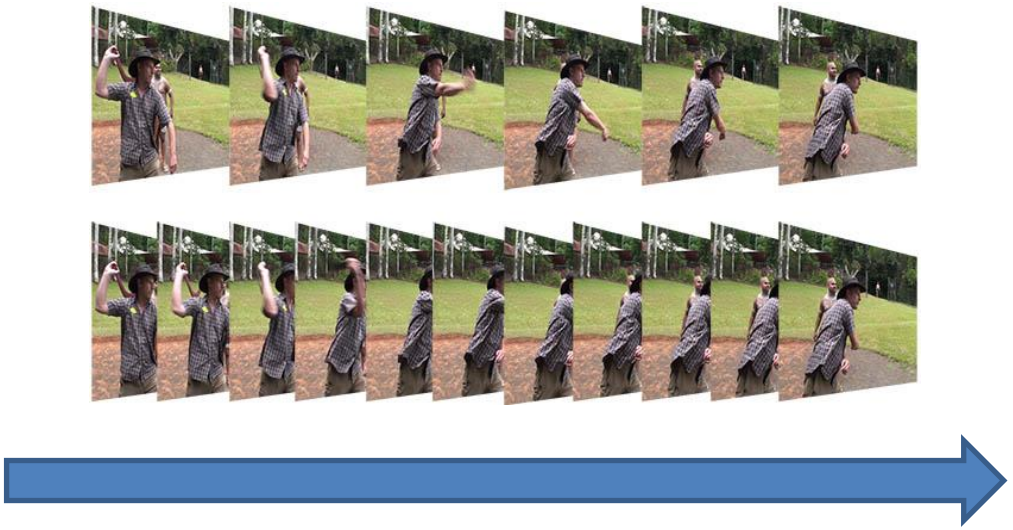
ICCV 2019

Proposal

- Self supervised representation learning in videos.
- Use representations learnt through self supervised learning on downstream tasks.

Self-Supervised Learning in Videos

- Most common self-supervised learning of videos is next frame prediction.
- Inspired from distributed learning of words in NLP (word2vec).

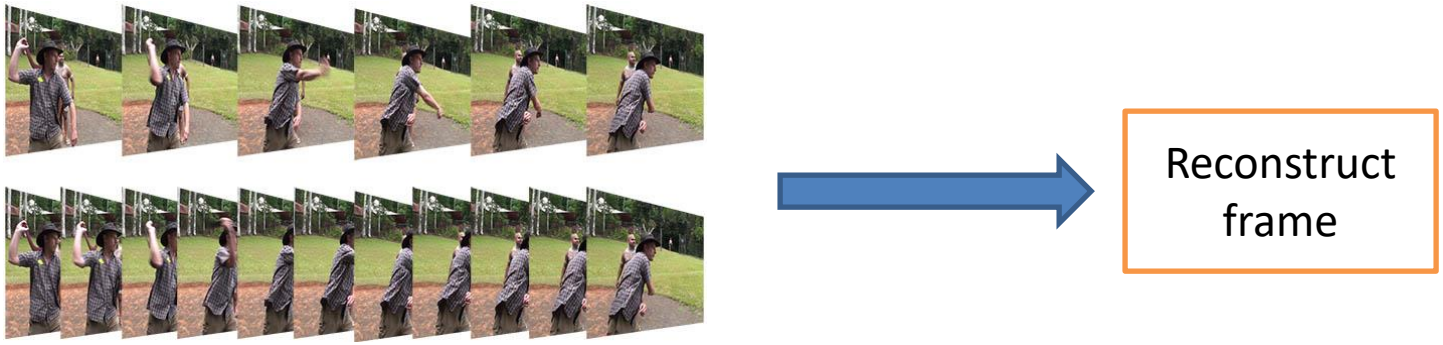


Next Frame?

Given a set of previous frames, predict future frames.

Previous Works

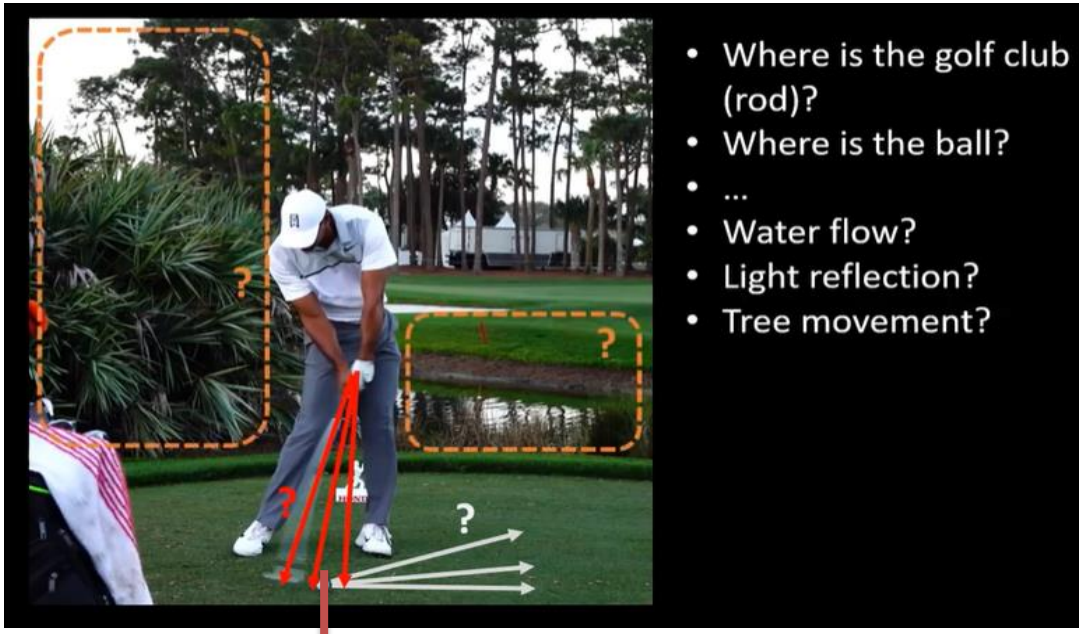
- Previous works on future frame prediction can be categorized into two approaches:
  - Predict a reconstruction of the actual frames.
  - Predict latent representations (embeddings) of the frames.



First Approach

Motivation

- Why reconstructing future frame is difficult?

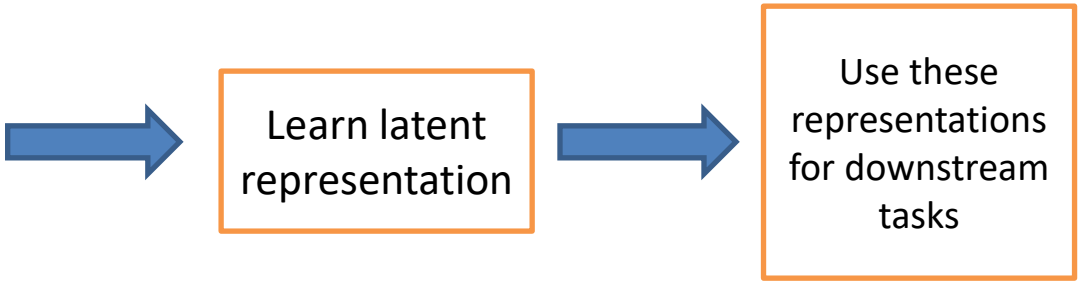


- Future is uncertain (Many possibilities).
- Too many factors affect the outcome such as brightness, appearance.

https://www.youtube.com/watch?v=43KIHUvHjB0

Motivation

- Instead of reconstructing future frame in self-supervised learning of videos, learn latent representations.
- If the end-goal is to use these representations on downstream tasks, this is even more beneficial.



Second Approach

Proposed Approach

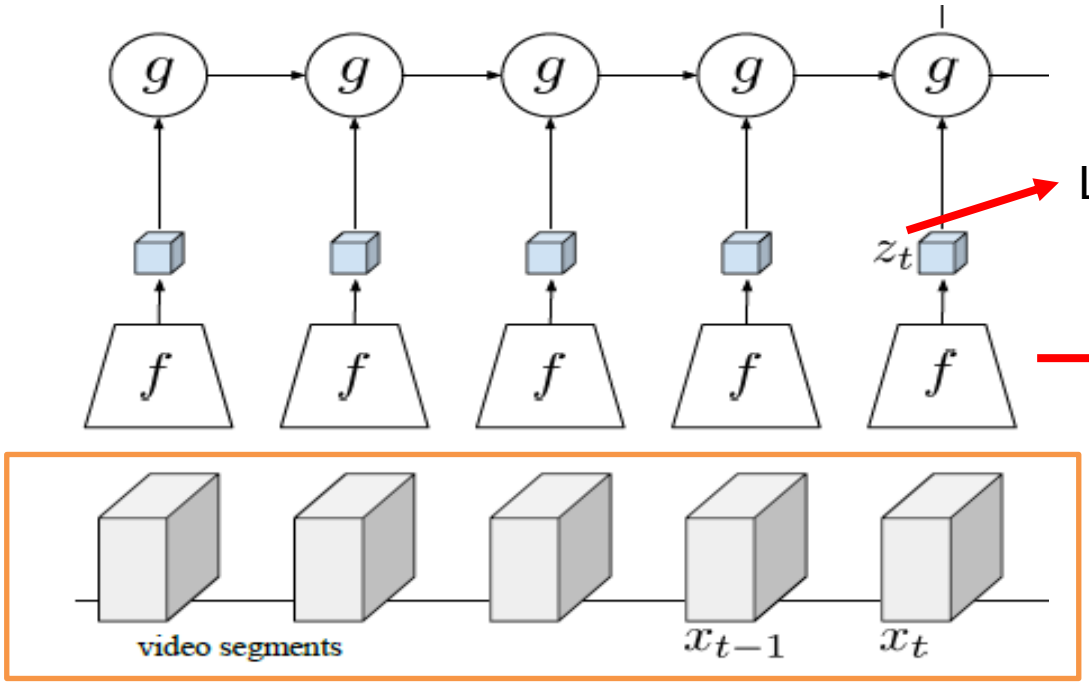
- Self-supervised representation learning on videos.
- Predict future frame representation.
- Fine-tune on downstream tasks using learned video representations.

Background

- Curriculum learning
  - Train machine learning models with increasing difficulty of samples.
  - First train with easy samples, then hard samples.
  - Identifying which samples are easy or hard is the most challenging part in curriculum learning.

- Noise Contrastive Estimation (NCE)
  - Used predominantly in training auto-regressive language models.
  - Constructs a binary classification task where the classifier is fed with a “true” sample and ‘k’ “noise” samples.
  - Objective is to distinguish one “true” sample among ‘k’ “noise” samples.

Architecture



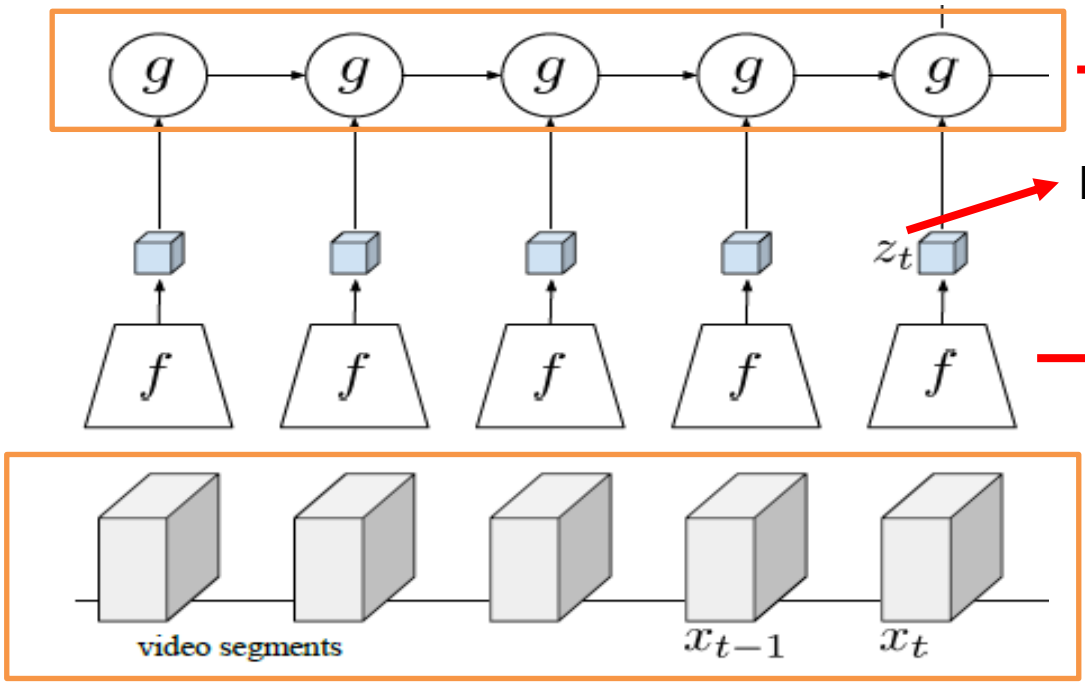
Latent representation

Non-linear encoder function to map videos into latent representations

Video clip partitioned into non-overlapping blocks

$$Z_t = f(x_t)$$

Architecture



Aggregation function for aggregating ‘t’ latent representations

Latent representation

Non-linear encoder function to map videos into latent representations

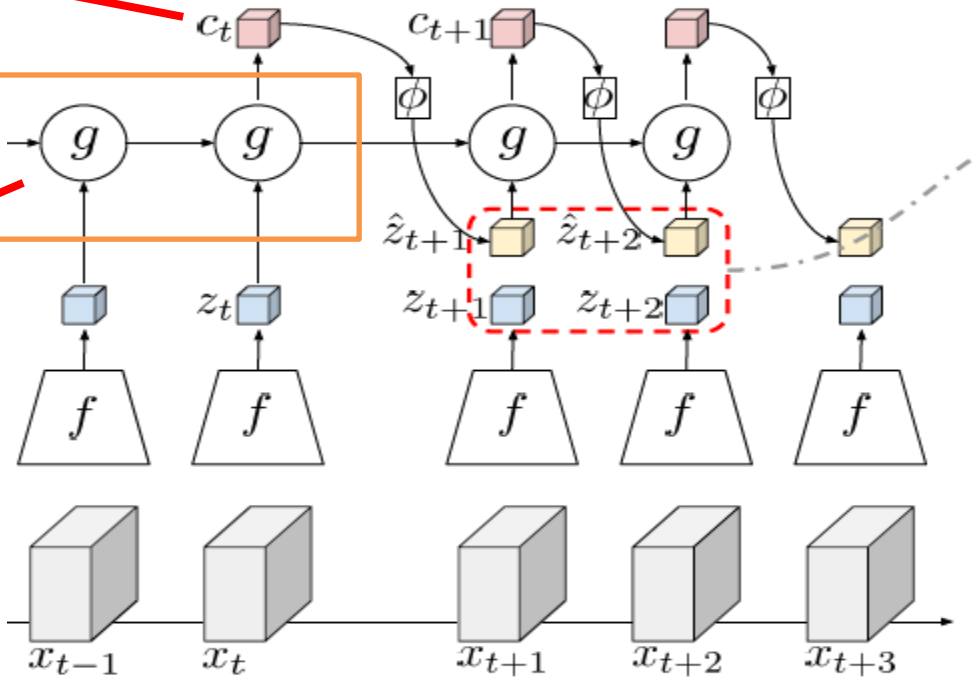
Video clip partitioned into non-overlapping blocks

Context representation: $c_t = g(z_1, z_2, \ldots, z_t)$

Architecture

Context representation of 't' past frames

Past frame representations

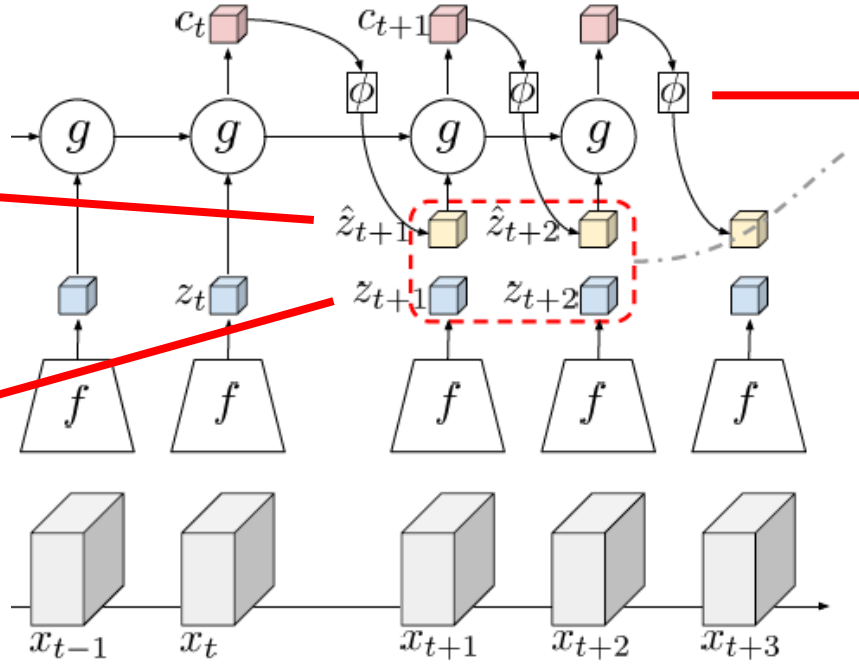


- Using 't' past frame representations predict 'q' future frame representations.
- Similar to Seq2Seq models in NLP.

Architecture

Future clip representation at 't+1' time step.

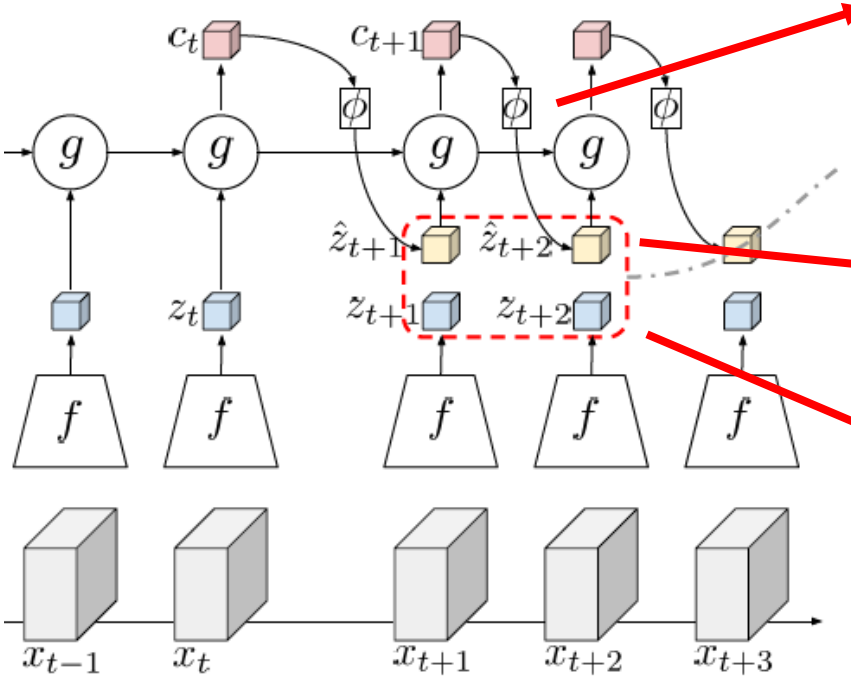
Ground truth frame representation at 't+1'



Predictor function

$$Z^{\wedge}_{t+1} = \phi(c_t) = \phi(g(z_1, z_2, \ldots, z_t))$$

Architecture



For prediction at time step ‘t+q’, predictor function considers previous predicted frame representation (t + q – 1) along with context representation.

Predicted future clip representation at ‘t+2’ time step.

Ground truth frame representation at ‘t+2’

$$\hat{Z}_{t+2} = \phi(c_{t+1}) = \phi(g(z_1, z_2, \ldots, z_t, \hat{Z}_{t+1}))$$

Training

- The ground truth representation z and the predicted representation z^ are computed.
- The representation for the i-th time step is denoted as z_i and $z_i^{\wedge}$
- The feature vector in each spatial location of the feature map is $z_{i,k}$ where 'i' is the temporal index and 'k' is the spatial index and k = {(1; 1); (1; 2); .., (H;W)}.
- The objective is to optimize:

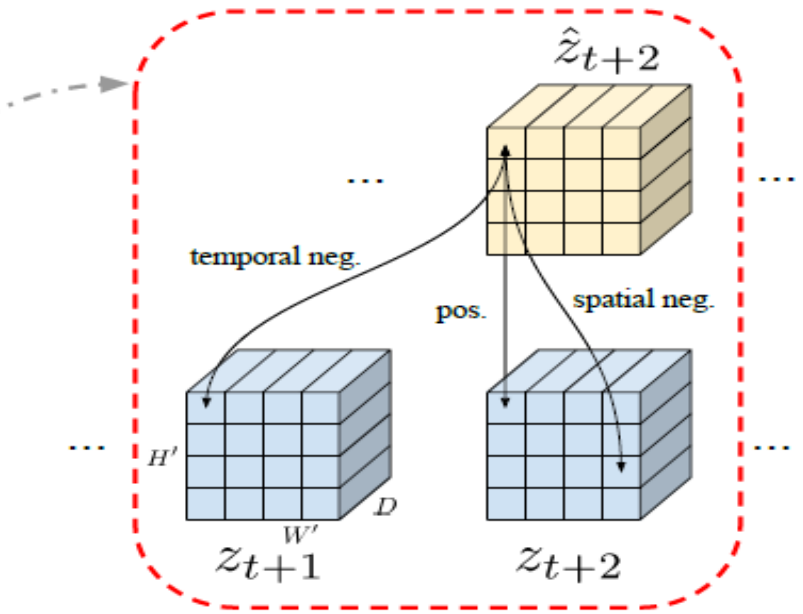
$$\mathcal{L} = -\sum_{i,k}\left[\log\frac{\exp(\hat{z}_{i,k}^{\top}\cdot z_{i,k})}{\sum_{j,m}\exp(\hat{z}_{i,k}^{\top}\cdot z_{j,m})}\right]$$

Predicted representation

Ground truth representation

Decides the quality of representations

Training



- For a predicted feature vector $z\hat{}_{i;k}$, the only positive pair is ($z\hat{}_{i;k}$; $z_{i;k}$) and rest are negative pairs when (i,k) != (j,m).
- The loss encourages the positive pair to have a higher similarity than any negative pairs.

Training

- Negative pairs
  - Easy negatives
    - Pred and ground truth are from different videos.
    - Easy pairs as they have different colour distributions.
  - Spatial negatives
    - Same video, but at a different spatial position.
    - Combatively more similarity between pred and GT
  - Temporal negatives
    - Same video, same spatial position but at a different temporal position.
    - Hardest pairs to classify as the pairs will have greater similarity.

- Use curriculum learning to gradually increase the number of hard negatives during training.
- Training with more hard negatives makes the representations more robust.

Why can't we train using hard negatives from the beginning then?

Architecture details

module	specification	output sizes $T \times t \times d^2 \times C$
input data	-	$5 \times (5 \times 128^2 \times 3)$
$f(.)$	see Table 7	$5 \times (1 \times 4^2 \times 256)$ (z)
$g(.)$	see Table 8	$1 \times 1 \times 4^2 \times 256$ (c)
$\phi(.)$	2-layer FC	$1 \times 1 \times 4^2 \times 256$ $(\hat{z})$
	compute loss using z and $\hat{z}$	

Table 6: The structure of DPC model.

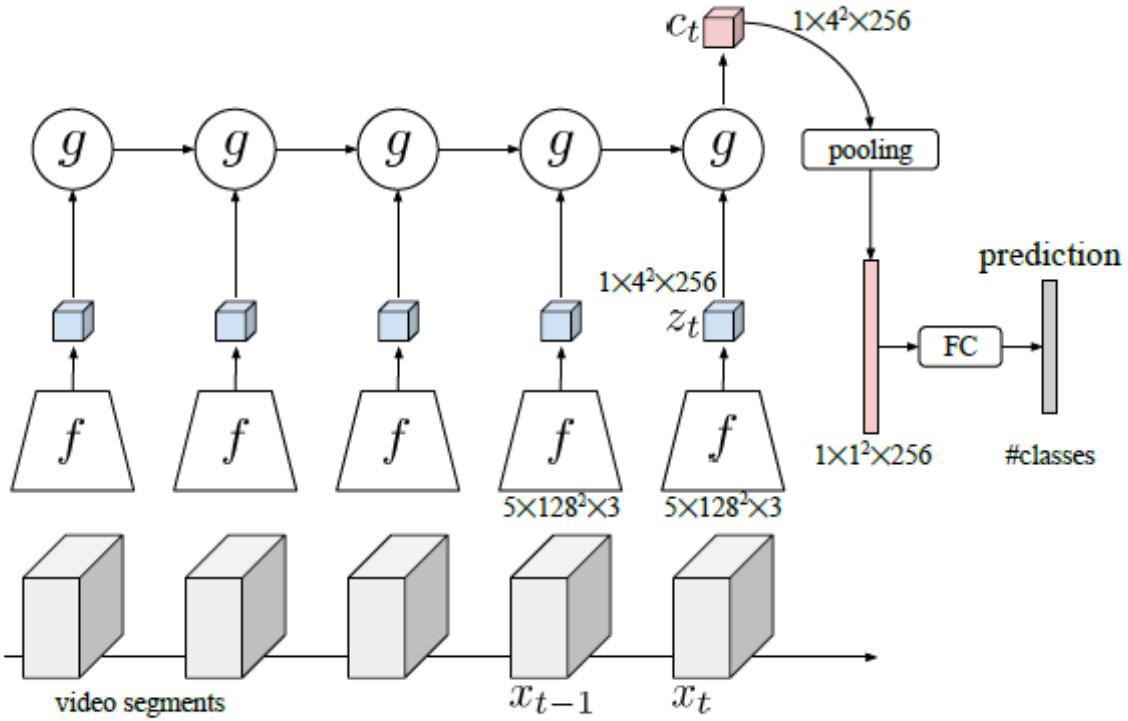
stage	specification	output sizes $T \times t \times d^2 \times C$
input data	-	$5 \times 1 \times 4^2 \times 256$
ConvGRU	$[1^2, 256] \times 1$ layer	$1 \times 1 \times 4^2 \times 256$

Table 8: The structure of aggregation function $g(.)$.

stage	specification	output sizes $t \times d^2 \times C$
input data	-	$5 \times 128^2 \times 3$
conv_1	$1 \times 7^2, 64$ stride $1, 2^2$	$5 \times 64^2 \times 64$
pool_1	$1 \times 3^2, 64$ stride $1, 2^2$	$5 \times 32^2 \times 64$
res_2	$\begin{bmatrix} 1 \times 3^2, 64 \\ 1 \times 3^2, 64 \end{bmatrix} \times 2$	$5 \times 32^2 \times 64$
res_3	$\begin{bmatrix} 1 \times 3^2, 128 \\ 1 \times 3^2, 128 \end{bmatrix} \times 2$	$5 \times 16^2 \times 128$
res_4	$\begin{bmatrix} 3 \times 3^2, 256 \\ 3 \times 3^2, 256 \end{bmatrix} \times 2$	$3 \times 8^2 \times 256$
res_5	$\begin{bmatrix} 3 \times 3^2, 256 \\ 3 \times 3^2, 256 \end{bmatrix} \times 2$	$2 \times 4^2 \times 256$
pool_2	$2 \times 1^2, 256$ stride $1, 1^2$	$1 \times 4^2 \times 256$

Table 7: The structure of the encoding function $f(.)$ with 3D-ResNet18 backbone.

Architecture details



Action classification architecture

Experiments

- An ablation study on the DPC model to show the function of different design choices.
- The benefits of training on a larger, and more diverse dataset.
- The correlation between performance on self-supervised learning and performance on the downstream supervised learning task.
- The variation in the learnt representations when predicting further into the future.

Experiments

- Datasets:
  - UCF101 (13k videos, 101 human action classes)
  - HMDB51 (7K videos from 51 human action classes)
  - Kinetics-400 (306K video clips for 400 human action classes).

- Evaluation Methodology:
  - The self-supervised model is trained either on UCF101 or K400. The predictive task is initially designed to observe the first 5 blocks and predict the remaining 3 blocks (5pred3).
  - The representation is evaluated by its performance on a downstream task action classification.
  - Top1 accuracy for evaluating self-supervised video representations and on downstream tasks.

Results

Ablation on architecture:

Network	setting	Self-Sup. (UCF) method	top1 acc	Sup. (UCF) top1 acc
R-18	-	- (rand. init.)	-	46.5
R-18	5pred3	DPC	**53.6**	**60.6**
R-18	5pred3	remove Seq.	51.3	56.9
R-18	5pred3	remove Map	36.5	44.9

- Full model architecture outperformed all the other variants.

Results

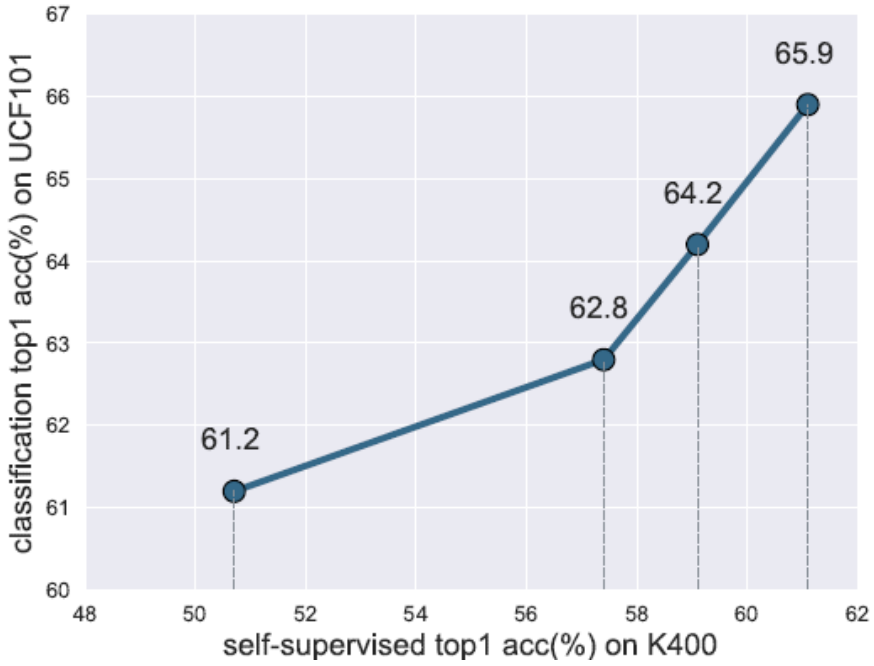
Benefits of using large datasets:

Network	setting	Self-Sup. dataset	top1 acc	Sup. (UCF) top1 acc
R-18	5pred3	UCF101	53.6	60.6
R-18	5pred3	K400	**61.1**	**65.9**

- Significant improvement when trained with kinetics dataset on both self-supervised accuracy and on action classification task.

Results

Self-Supervised vs. Classification Accuracy



- As the self-supervised performance increases, downstream accuracy also increases.

Results

Benefits of Predicting Further into the Future

Network	Self-Sup. (K400)			Sup. (UCF)
	setting	curr.	top1 acc	top1 acc
R-18	5pred3	✗	61.1	65.9
R-18	4pred4	✗	48.3	64.9
R-18	5pred3+4pred4	✓	**50.8**	**68.2**

Curriculum training on 4pred4 achieved 2.3% performance boost on downstream task compared to 4pred4 trained from scratch.

Results

Comparison with State-of-the-art 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**

This method outperformed all the other self-supervised techniques on action recognition task.

Results

Visualization of nearest neighbours

(a)

(b)

a: Cosine similarity among videos using DPC representations.
b: Inflated image net pretrained weights

Contributions

- Different approach for self-supervised video representation.
- Outperformed other self-supervised models.

Weaknesses

- Approach is not so novel.
- Ablation studies are ‘weak’.
- Not so significant qualitative results.
- Difficult to understand how good the model is.

Discussion

- What do you think are the advantages/disadvantages of using single context representation for all the future predictions?

- How to best evaluate these types of self-supervised representation models?

Questions?

Thank you