Knowledge-Based Systems 210 (2020) 106534

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Self-supervised representation learning by predicting visual permutations

Qilu Zhao, Junyu Dong *

Department of Computer Science and Technology, Ocean University of China, No. 238 Songling Road, Laoshan District, Qingdao, China

ARTICLE INFO

Article history:
Received 18 October 2019
Received in revised form 26 September 2020
Accepted 13 October 2020
Available online 14 October 2020

Keywords:
Unsupervised representation learning
Self-supervised learning
Jigsaw puzzle reassembly
Multi-task learning
Permutation prediction

ABSTRACT

We propose a self-supervised learning method to uncover the spatial or temporal structure of visual data by identifying the position of a patch within an image or the position of a video frame over time, which is related to *Jigsaw puzzle reassembly problem* in previous works. A Jigsaw puzzle can be seen as a shuffled sequence, which is generated by shuffling image patches or video frames according to an unknown permutation. The task of predicting the visual permutations can be used to train a learning system to capture structural information which is important for semantic-level tasks, such as object recognition and action recognition. To this end, we propose a multi-task learning framework where a group of principal tasks aims to predict the index of each sample in the original sequence, and a group of auxiliary tasks aims to predict the spatial or temporal relation of adjacent samples in the shuffled sequence. Our scheme can handle the whole space of permutations and is fairly scalable, and it is also generic to solve many problems such as self-supervised representation learning, relative attributes, and learning to rank. Our method achieves state-of-the-art performance on the STL-10 benchmarks for unsupervised representation learning, and it is competitive with state-of-the-art performance on UCF-101 and HMDB-51 as a pretraining method for action recognition. In addition, we apply the proposed method on age comparison task to prove it is generic to solve ranking problems.

1. Introduction

Self-supervised learning has opened an intriguing new avenue into unsupervised learning. Especially, it has emerged as the new state-of-the-art learning framework for unsupervised representation learning [1–4]. The basic idea for all self-supervised learning methods is to formulate a pretext task with automatically generated supervisory signals [5]. For visual understanding, these pretext tasks must be designed in such a way that high-level image understanding is useful for solving them [3]. Thus, the intermediate layers of the trained neural networks are able to encode high-level semantic representations which are useful for solving the downstream tasks [6–8], e.g. object detection, semantic segmentation, and so on. For example, Gidaris et al. [9] propose to generate 4 copies of a single image by rotating it by $\{0°, 90°, 180°, 270°\}$ and let a convolutional neural network (CNN) [10] to recognize the geometric transformation that is applied to the image. The rotation recognization is transformed into a classification task by regarding the degree as category. Intuitively, solving this task will require to understand the concept of the objects depicted in the image.

In this paper, we focus on exploiting spatial and temporal structure of visual data for semantic-level feature learning with self-supervised learning methods. Inspired by the *Jigsaw puzzle reassembly* problem as shown in Fig. 1, we design a ranking model that is capable of reordering a shuffled sequence. Given an ordered sequence, a training sample is generated by shuffling it according to a randomly determined permutation, and we can leverage the index of each image in the original sequence and the structural relation of adjacent images in the shuffled sequence as the supervision signals. Solving the sequential recovery task requires the model to learn the spatial or temporal structure of visual data. Thus, the learned model can be used to solve some downstream tasks, including object recognition, action recognition, and the relative attributes task. For example, we can directly use the trained model as a feature extractor following the paradigm in [4], and the obtained feature can be applied to image classification and object detection. Generally, self-supervised learning aims to provide a general-purpose pre-trained model for other tasks. Previous work [11] identified that using even a few early layers from a pre-trained ImageNet model can improve both the speed of training, and final accuracy, of visual models. However, there are no pre-trained models in some domain. For instance, there are very few pre-trained models in the field of medical imaging. To solve this problem, our method is

* Corresponding author.
E-mail addresses: zql@ouc.edu.cn (Q. Zhao), dongjunyu@ouc.edu.cn (J. Dong).

https://doi.org/10.1016/j.knosys.2020.106534
0950-7051/

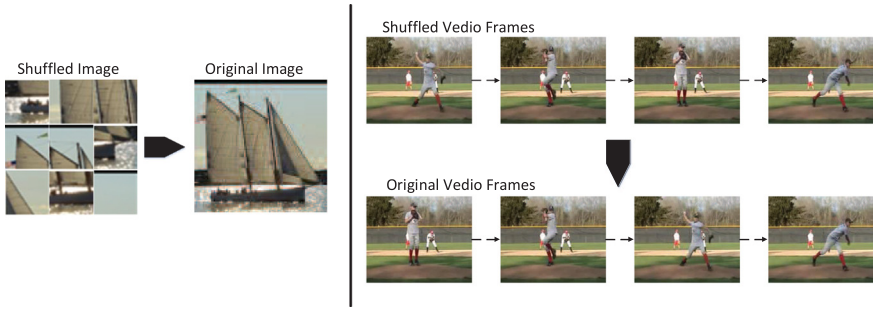

Fig. 1. Illustration of the *Jigsaw puzzle reassembly* problem. It originally refers to the spatial layout recovery problem (left), and we generalize it to the reordering of shuffled video frames (right). Our goal is to learn visual features by solving the recovery problem.

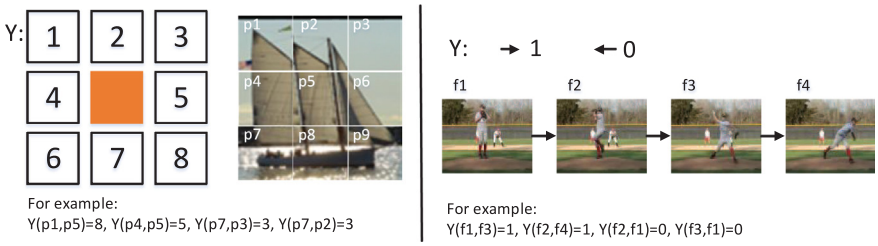

Fig. 2. Illustration of the relation labeling. One of the eight possible spatial arrangements is considered as the relation label for a pair of patches (left). For a pair of video frames, the correct temporal order is labeled 1, otherwise 0.

able to generate a pre-trained model for medical imaging models, such as segmentation, classification, and so on.

The most relevant previous work is the Jigsaw puzzle solver [5], that focuses on the spatial layout recovery problem. Jigsaw puzzle solver exploits a tiny subset of possible permutations to shuffle the image sequences (only 100 permutations from 362,000 possible permutations), and it formulates the spatial layout recovery problem as a 100-class classification task by regarding the permutation as category. In order to handle the whole space of permutations, we propose a multi-task learning framework to solve the sequential recovery task, which is fairly scalable and generic to many downstream tasks. The framework we propose consists of two groups of surrogate tasks. A group of principal tasks aims to predict the index of each sample in the original sequence, and a group of auxiliary tasks aims to predict the structural relation of adjacent samples in the shuffled sequence. The first group can complete the recovery task by itself, but it does not exploit the relatedness of the principal tasks explicitly, resulting in the lack of discriminative feature that represents relative relationships between samples. Thus we leverage the second group to enhance the learning of relations, which will eventually enhance the discriminative power of the learned features. We build a multi-stream convolutional neural networks (MS-CNN) that receives a shuffled sequence of samples (image patches or video frames) as input, and we apply it to learn spatial or temporal representation depending on the structural relation. According to the input data, the structural relation is relative spatial location or temporal order. The labeling method is shown in Fig. 2. For image patches, these surrogate tasks require the learning system to extract objects and their parts in order to reason about their relative spatial configuration. For video frames, these surrogate tasks require reasoning about object transformations and relative locations through time, which in turn forces the representation to capture object appearances and deformations.

Our main contributions are:

1. Our method can handle the whole space of permutations, which significantly improves the generalization ability of the learned network.
2. Our method can apply to various kinds of image sequences, such as image patches, video frames, facial images labeled by ages, and other image sequences ordered by a certain attribute. This innovation makes it generic to solve many downstream tasks, such as object recognition, action recognition, age comparison, and so on.
3. The designed network architecture is flexible and extensible for the model to deal with sequences of an arbitrary length.
4. Experimental results involving several downstream tasks have confirmed the superior performance of our method. It outperforms the state-of-the-art results for unsupervised representation learning on STL-10 dataset, and it achieves competitive performance for action recognition as a pre-training method on UCF-101 and HMDB-51.

2. Related work

Self-supervised learning can be considered as a branch of unsupervised learning that aims to use data without any annotation. Traditional unsupervised learning methods include dictionary learning [12], independent component analysis [13], auto-encoders [14], matrix factorization [15], and various forms of clustering [16,17]. Self-supervised learning leverages intrinsic reward signals for pretext tasks to learn general-purpose features. The pretext tasks are also known as self-supervision tasks because the signals are created automatically from the data.

Being a generic framework, many self-supervised tasks have been proposed for a wide number of applications in recent years, ranging from robotics to visual understanding. In robotics, self-supervised signals come from sensory inputs or the result of interacting with the world [18,19]. In visual understanding, self-supervision tasks generally involve taking a complex signal, hiding part of the input, and then requiring the model to predict the missing part. The tasks can broadly be divided into those that use auxiliary information or those that only use raw pixels.

Many works use auxiliary information such as multi-modal information [20]. Examples include: exploring the utility of ego-motion as supervision [21], predicting sound given videos [22], or

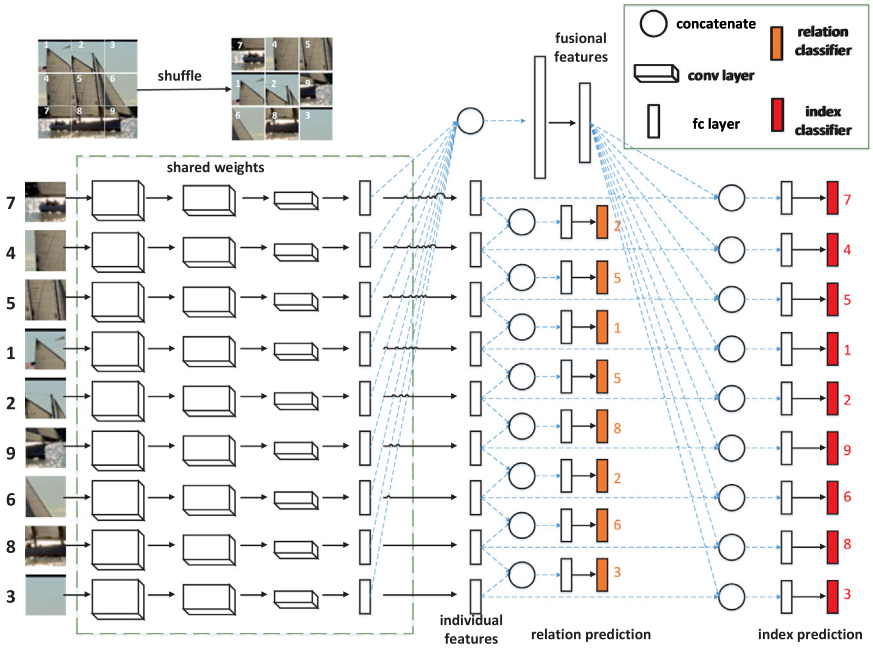

Fig. 3. Network architecture. We adopt a multi-stream CNN for the proposed framework. It receives a shuffled sequence of visual samples as input. Each sample in the sequence goes through a different branch. The first few layers share parameters, so do classifiers for the same task. The dotted lines denote the data flow of concatenating operations. Classification layers predict the index and relation with fused representation. The relation is labeled as shown in Fig. 2. For example, patch 4 is at the top of patch 7, thus the relation label is 2 for patch pair (7,4).

predicting what robotic motion caused a change in a scene [23]. However, auxiliary information like non-visual information is still difficult to obtain, thus many works focus on predicting missing raw pixels from input parts. Examples include image inpainting [24], image colorization [25], its improved variant [26], and motion segmentation prediction [27]. Another approach generates cleverly designed image-level classification tasks by creating class labels from raw pixels. RotNet [9] proposed to predict rotation angle of a rotated image. DeepCluster [28] considered clustering of the images as categories. Exemplar [4] created pseudo-classes by augmenting each original sample via translation, rotation, scaling, and color shifts. A network was trained to discriminate between pseudo-classes. Context prediction [29] proposed a successful pretext task of predicting the spatial configuration for a pair of image patches. This work spawned a line of work that is related to sequential image processing [30], including Jigsaw puzzle solver [5] and its follow-ups [31,32]. Video jigsaw [33] and sequence sorting [34] extended the method of jigsaw puzzle solver to video representation learning problem, individually. DeepPermNet [6] introduced Sinkhorn layer for permutation matrices prediction to improve Jigsaw puzzle solver, allowing their method to handle the full set of permutations. Besides, exploring temporal order from videos [35,36] is also a good idea to design unit-level classification tasks.

3. Method

In this section, we describe our method for self-supervised representation learning. We first introduce the multi-task learning framework. Then, we describe network architecture and data sampling strategies.

3.1. Multi-task learning framework

Given an ordered sequence of samples (image patches or video frames), we generate shuffled sequences by randomly shuffling. The goal of this work is to recover the original sequences by predicting the permutations of shuffled ones. We hypothesize that the models trained by solving this task are able to learn semantic-level representations which capture high-level semantic concepts, structure, and visual patterns.

We cast the recovery task into a series of index prediction tasks. Suppose we have an ordered sequence of images $X_o = \langle I_1, I_2, \ldots, I_n \rangle$, we shuffle it randomly to get a disordered sequence, for example $X_s = \langle I_j, I_k, \ldots, I_n, I_2 \rangle, j < k < n$. Correspondingly, we need to solve n index prediction tasks by learning a parametrized function $f_\theta : I \rightarrow L$, where L denotes the index set, and θ denotes the parameters. For the previous example, $f_\theta(I_j) = j$ and $f_\theta(I_k) = k$. Regarding the index as category label, we formulate each index prediction task as a classification task. Similar learning schemes proved successful in previous works [5,29,34]. Completing the index prediction tasks can solve the recovery problem, thus we consider them the principal tasks in the proposed multi-task learning framework. However, these tasks does not take advantage of the relatedness between them explicitly, which results in lack of some important information of relative structure. Thus we propose a group of auxiliary tasks to enhance the learning of relations, which will eventually enhance the discriminative power of the learned features.

The auxiliary tasks aim to predict the structural relation (spatial or temporal relation) of adjacent samples in the shuffled sequence. Suppose the previous example $X_s = \langle I_j, I_k, \ldots, I_n, I_2 \rangle$ is a sequence of video frames, we need to solve $n-1$ relation prediction tasks by learning a parametrized function $\Gamma_\theta : I \times I \rightarrow Y$, where Y is the set of relation labels. The labeling method is shown in Fig. 2. For the previous example, $\Gamma_\theta(I_j, I_k) = 1$ and $\Gamma_\theta(I_n, I_2) = 0$. Similar to index prediction task, we also formulate relation prediction task as classification. Formally, we optimize the following problem in our framework:

$$\min_\theta \sum_{i=1}^{n} \Delta(f_\theta^i(I), L) + \lambda \sum_{i=1}^{n-1} \Delta(\Gamma_\theta^i(I \times I), Y), \tag{1}$$

where $\Delta(\cdot, \cdot)$ is a loss function, i indexes the branch of the multi-stream architecture, and λ is a weighting hyperparameter.

3.2. Network architecture

We exploit a multi-stream CNN to implement the proposed framework as shown in Fig. 3. The network consists of 3 types of modules: siamese module including the first few layers, feature fusion module, and classification module. Siamese module sharing parameters across different branches aim to extract initial features for each sample. Feature fusion module is the key part. It consists of concatenation, convolutional layers and fully-connected layers. For the index prediction, it first concatenates all initial features from siamese module, and learns the fusional feature with convolutional/fully-connected layers. Then it concatenates the fusional feature with the individual feature of each sample to form a final representation for the index classifier. For the relation prediction, it concatenates individual features of adjacent samples to form a final representation for the relation classifier. The feature fusion module aims to get a global view representing the whole sequence of samples and adjacent samples. The index classifiers need to "see" both the global view and local view for identifying the position of each sample in the sequence. Classification module consists of the index classifier and relation classifier. Each classifier is implemented by a softmax layer. Thus, $\Delta(\cdot,\cdot)$ in Eq. (1) adopts the cross entropy loss.

3.3. Training data sampling

Data sampling strategy is critical for self-supervised learning. It is important to avoid the network using low-level cues to complete the task. We exploit several training tricks to deal with this problem. While exploring the spatial structure, we first divide each image into a 3×3 grid, and then we shuffle them randomly to get a sequence of image patches. Three sampling strategies are leveraged to force the network to extract high-level semantics. First, we cut the central region out of each patch generating a gap between patches, which can prevent boundary patterns or textures from continuing between patches [29]. Second, we randomly flip the cropped patches vertically and horizontally, which proved an effective data augmentation strategy [37]. Third, we randomly choose one color channel and duplicate the values to other two channels, which imposes additional challenges for the network [34].

While dealing with video frames, we take advantage of one more strategy to avoid "trivial" learning, besides the 3 strategies mentioned above. We observe that it is almost impossible for the network to order the shuffled frames sampled in temporal windows with very little motion. To avoid generating ambiguous training examples, we treat the magnitude of optical flow per-frame as a weight for that frame, and use it to select high motion windows [35]. In addition to frame selection, we further extract spatial patches of high motion within the selected frames [34], resulting in 9 patches for each sequence.

We use facial images to perform the age comparison task. Considering the age range is large, we split the facial images into several overlapped groups. We sample two kinds of sequence: inter-group sequence and inner-group sequence. Each sample in a inter-group sequence is from a different group, and samples in a inner-group sequence are all from a same group. No two samples in a sequence are labeled with the same age. Before feeding the samples to the multi-stream networks, we adopt a general pre-processing procedure for face detection and alignment. We first leverage Harr-based cascade classifiers [38] to detect the face. Then, we align the face based on the locations of eyes. Finally, the image is resized to a standard size of $112 \times 112 \times 3$ for training and testing.

4. Experiments

In this section, we validate the effectiveness of our method for self-supervised representation learning using two semantic-level tasks, image classification and action recognition. We treat the trained network as a feature extractor for image classification task. This evaluating scheme is more direct than the pre-training pipeline to prove the effectiveness of self-supervised methods, but we apply the trained network as a pre-trained model for action recognition, which is convenient to compare with previous works. In addition, we leverage the age comparison task to prove our method is generic to solve ranking problems.

4.1. Datasets

We evaluate our method on STL-10 [39], UCF-101 [40], HMDB-51 [41], and MORPH2 [42]. STL-10 contains 96×96 pixel images of 10 classes (500 training images and 800 test images per class) and extra 100,000 unlabeled images. It is especially well suited for unsupervised representation learning as it contains a large set of unlabeled samples. In the experiments, we trained our model from the unlabeled subset of STL-10. The training procedure was repeated 3 times, and we report the average performances.

UCF-101 is a famous benchmark video dataset for action recognition. It consists of 101 action categories and 13,320 clips with about 9.5k for training and 3.5k for testing. HMDB-51 consists of 51 action categories with about 3.4k clips for training and 1.4k clips for testing. We report results using split 1 of both UCF-101 and HMDB-51 datasets. Classification accuracy is the standard metric for action recognition performance on these datasets. We trained our model on the training split of UCF-101, and used it to finetune on both datasets for action recognition.

MORPH2 [42] is a popular benchmark dataset for facial age estimation. It consists of about 55,000 facial images of 13,000 people, whose ages range from 16 to 77 years old. We randomly divide the dataset into independent training (80%) and testing (20%) sets, where the facial images are split into 5 overlapped groups: [-,25], [20,35], [30,45], [40,55] and [50,-]. Sequences are randomly sampled in a 1-to -5 ratio of inter-group to inner-group. The length is 5. We conducted experiments of age comparison on this dataset.

4.2. Implementation details

We implement our method on TensorFlow [57] with two GTX 1080Ti. While dealing with different types of data, we use different network architectures with the same framework shown in Fig. 3. For comparison with previous works, we use a slight modification of AlexNet (M-AlexNet) for action recognition in the experiment. We adopt another Convnet architecture, named PVP-Net, for representation learning and age comparison, because the kernel size of the first convolutional layer in AlexNet is too large for $96 * 96$ images in STL-10 and $112 * 112$ images in MORPH2. The Convnet architectures are shown below:

- PVP-Net: shared weights {C128(5)-C96(3)-P-C96(3)-C96(3)-C96(3)-P-C96(3)-C96(3)-P-C64(3)-C64(3)}, fusional features {C256(3)-F4096}, individual features {F1024}, relation prediction {Softmax}, and index prediction {Softmax}.
- M-AlexNet: shared weights {C96(11)-P-C256(5)-P-C512(5)-P-C384(3)-C384(3)-C256(3)-P-F512}, fusional features {F2048}, individual features {F2048}, relation prediction {Softmax}, and index prediction {Softmax}.

Table 1
Classification results on STL-10.

Method	Acc (%)	Method	Acc (%)
SWWAE [43]	74.33	VAE [44]	68.65
IIC [45]	79.2	β-VAE [46]	70.53
AAE [47]	64.15	BiGAN [48]	74.77
NAT [49]	70.55	DIM [50]	78.21
CPC [2]	77.81	ADC [51]	56.7
DeepCluster [28]	73.4	Conv. Clustering [52]	74.1
Target Coding [53]	73.15	Exemplar-CNN [4]	74.2 ± 0.4
Jigsaw puzzle solver [5]	76.62 ± 0.17	DeepPermNet [6]	74.71 ± 0.22
Ours (index)	77.76 ± 0.19	Ours (index + relation)	79.58 ± 0.26

Table 2
Finetuning results on UCF-101 and HMDB-51.

Pretraining	Acc on UCF-101 (%)	Acc on HMDB-51 (%)
Random	40.0	16.3
ImageNet (with labels)	67.7	28.0
Invariant mapping [54]	45.7	16.3
Scene dynamics [55]	52.1	–
Trajectories [56]	40.7	15.6
Sorting [34]	56.3	22.1
Shuffle-and-learn [35]	50.9	19.8
Odd-One-Out [36]	60.3	32.5
Video Jigsaw [33]	55.4	27.0
Jigsaw puzzle solver [5]	57.5	24.4
DeepPermNet [6]	58.3	23.9
Ours (index)	58.8	25.1
Ours (index + relation)	59.6	25.7

The abbreviations Ck(s), Fk, P represent a convolutional (C) layer with k kernels of size $s * s$, a fully-connected (F) layer with k filters, and max-pooling (P) layers respectively. Max-pooling is performed over a 2×2 pixel window, with stride 2. The stride of all convolutional layers is 1 pixel, except the first layer in M-AlexNet (2 pixels). We use ReLU non-linearity (ReLU) [58] after every convolutional/fully-connected layer. Batch normalization (BN) [59] is adopted after each convolutional layer before ReLU. BN and ReLU are omitted in the denotations above.

No pre-processing is applied to training images except ZCA whitening. For each image in STL-10, we split it into a 3×3 grid and cut the $28 * 28$ central region out of each patch, leaving a gap with 8 pixels between patches. For each selected frame in UCF-101 and HMDB-51, we first resize it to $160 * 120$ and extract $100 * 100$ patches of high motion, then we randomly cut a local region out of each patch instead of extracting the central region, resulting $80*80$ patches as inputs for training. All weights are initialized from a zero-centered Normal distribution with standard deviation 0.02. All models are trained with mini-batch Adaptive Moment Estimation (Adam) [60] with a mini-batch size of 50 unless otherwise noted. For training on STL-10, we use a learning rate that is 0.001 for the first 100,000 steps, 0.0005 for steps 100,001 to 110,000, and 0.0001 for the remaining 130,000 steps. For training on UCF-101, the sampling strategies result in around 600,000 sequences, and we use a learning rate that is 0.0005 for the first 10 epoches, 0.0001 for the next 10 epoches, 0.00005 for the next 20 epoches, and 0.00001 for the remaining 20 epoches. For training on MORPH2, we use a learning rate that is 0.001 for the first 50,000 steps, 0.0005 for steps 50,001 to 100,000, and 0.0001 for the remaining 50,000 steps. The hyperparameter λ is set to 1.0 empirically for all models implemented in the experiments.

4.3. Classification results on STL-10

While evaluating the trained network for representation learning on STL-10, we apply it as a generic feature extractor on $96*96$ images and leverage a image classification task for testing. Internal features are extracted from feature maps of convolutional layers in the shared parameters module of PVP-Net. We use the max-pooling operation to extract features, resulting in 16 values per feature map. The pooled features are flattened into vectors, and we concatenate them to form one unified representation of the image. After feature extraction, we train a softmax classifier without regularization for the image classification task on the training set of STL-10 with 5,000 images. The performance is shown in Table 1.

We compare our method against various unsupervised/self-supervised methods: Variational AutoEncoders (VAE [44]), β-VAE [46], Adversarial AutoEncoders (AAE [47]), SWWAE [43], BiGAN [48], Noise As Targets (NAT [49]), Contrastive Predictive Coding (CPC [2]), Invariant Information Clustering (IIC [45]), Associative Deep Clustering (ADC [51]), DeepCluster [28], Exemplar-CNN [4], Deep InfoMax (DIM [50]), and so on. Auto-encoder and its variants are classic methods for unsupervised representation learning. Self-supervised learning methods continue to improve the state-of-the-art performance on STL-10, including CPC, NAT, Exemplar-CNN, Jigsaw puzzle solver, and so on. DIM and IIC are the latest works on self-supervised representation learning, and IIC set a new global state-of-the-art over all previous methods. All the experimental results of the competing methods shown in the top half of Table 1 are directly quoted from the related references. The bottom half of Table 1 shows the comparison of our method and closely related works, which all aim to recover the original sequence from shuffled ones. We implement Jigsaw puzzle solver [5], DeepPermNet [6] and our method with the similar architecture, similar training setting, and same evaluating scheme for fair comparison. As we can see, our method slightly outperforms the state-of-the-art method by leveraging two groups of auxiliary tasks, and the sequence recovery methods all outperform the classic auto-encoder variants on STL-10, which proves the effectiveness of this approach. Jigsaw puzzle solver has a deficiency that it can only exploit a tiny subset of possible permutations. Our method outperform Jigsaw puzzle solver by about 2.96%, and DeepPermNet cannot achieve the same performance as Jigsaw puzzle solver. Apparently, this observation shows that our method is more effective than DeepPermNet and

Table 3
Age comparison accuracy on MORPH2.

Group	#Sequences	DeepPermNet	Ours(index+relation)
[−, 25]	400	0.712 ± 0.002	0.766 ± 0.004
[20, 35]	800	0.740 ± 0.002	0.784 ± 0.005
[30, 45]	800	0.688 ± 0.008	0.716 ± 0.006
[40, 55]	600	0.813 ± 0.004	0.832 ± 0.002
[50, −]	200	0.852 ± 0.007	0.894 ± 0.006
inter-group	200	0.886 ± 0.001	0.920 ± 0.002

Table 4
Evaluation of the gap trick on STL-10.

Method	Acc (%)
Leaving gap (index)	77.76 ± 0.19
No gap (index)	73.28 ± 0.23
Leaving gap (index + relation)	79.58 ± 0.26
No gap (index + relation)	75.91 ± 0.22

Jigsaw puzzle solver on self-supervised representation learning. Another observation is that the auxiliary tasks do improve the performance of our method. It is obvious that the auxiliary tasks succeed in enhancing the discriminative power of the learned features.

4.4. Action recognition

For UCF-101, we uniformly sample 25 frames per video at test time, and the prediction for the video is an average of the predictions across these 25 inputs. For HMDB-51, we sample 1 frame per second from each video. All frames are resized to 160 × 120. After the network is trained on UCF-101, we use it as a pre-trained model for action recognition. We use another slight modification of AlexNet for finetuning, which shares the architectures of convolutional layers with M-AlexNet and fully-connected layers with standard AlexNet. We initialize it with the parameters of convolutional layers obtained from the pre-trained network, and finetune it for 40k steps with a batch size of 128, and learning rate of 0.01 decaying by 10 after 25k steps, using Adam and dropout of 0.8 after each fully-connected layer.

The results are shown in Table 2. Odd-One-Out uses stacks of frames differences as inputs, and other methods use static RGB images. Similar to the results on STL-10, our method show competitive performances. It almost achieves the best performance on UCF-101, which is satisfying, considering the Odd-One-Out method takes advantage of more temporal information during finetuning stage. Video Jigsaw uses spatial and temporal information together for the pre-training stage, resulting in a good performance on HMDB-51. Compared with the random initialization, our method significantly improves the performance of finetuned network. Meanwhile, we observe that the self-supervised methods still fall behind the supervised approach, but the gap reduces gradually.

4.5. Age comparison

We leverage a simple curriculum learning strategy [61] in the training procedure. The first 25,000 steps are performed on inter-group sequences, and the rest on inner-sequences. When testing, we first predict the order for sequences of facial images. Then we compute the pairwise accuracy for all pairs in each sequence. We randomly sample 3,000 sequences for testing, which results in 360,000 pairs. The permutation prediction is determined by the outputs of index classifiers with the maximum probability. In this experiment, we compare our method with DeepPermNet [6]. For fair comparison, we use similar architecture and similar training setting as possible as we can. The training procedure was repeated 3 times, and we report the average performances.

Table 5
Evaluation of the frame selection strategy on UCF-101 and HMDB-51.

Pretraining	UCF-101 (%)	HMDB-51 (%)
High motion(index)	58.8	25.1
Random sampling(index)	47.1	18.3
High motion(index + relation)	59.6	25.7
Random sampling(index + relation)	46.9	17.7

Table 6
The influence of sequence size.

Division	Acc on STL-10 (%)	Number of samples	UCF-101 (%)
2 × 2	71.50 ± 0.16	4	56.1
3 × 3	73.28 ± 0.23	9	59.6
3 × 4	72.11 ± 0.32	12	60.2
4 × 4	70.43 ± 0.10	16	58.5

As shown in Table 3, our method outperformed DeepPermNet [6] by an average margin of about 3.68% in pairwise accuracy. It is a substantial result, consistently observed across all groups. Apparently when the age differences in a sequence are large, both methods gained high accuracy, such as the inter-group sequence. The performances of both methods present a similar trend across the inner-groups. For example, it is easier to compare the age between old person than young, and range [30,45] is the most difficult group for the task.

4.6. Ablation study

In this section we discuss some critical design choices in detail, including the gap trick, frame selection strategy, sequence size, image size, and the feature fusion module.

The gap trick. This trick means we leave a gap between patches when extracting them from an image to form a sequence. When designing a pretext task, care must be taken to ensure that the task forces the network to extract the desired information without taking "trivial" shortcuts. In our case, low-level cues like boundary patterns or textures continuing between patches could potentially serve as such a shortcut, which would prevent the network from extracting the high-level semantics. Leaving a gap between sampled patches is a good solution to this problem. Following the evaluating scheme in Section 4.3, we implement a comparison experiment as shown in Table 4. Removing the gap trick would decrease the classification performance of the learned feature. These experimental results prove that the trivial shortcuts do happen when discarding the gap trick.

The frame selection strategy. In order to learn the temporal structure from video, we select several frames from a video clip to produce training sequences. Specifically, we take samples from high motion windows to avoid generating ambiguous training examples. It is an effective strategy as shown in Table 5. Random sampling degrades the performance drastically because the training samples contain similar frames that are impossible to be sorted.

The sequence size. The number of samples in the shuffled sequence is also critical for the self-supervised representation learning. In spatial structure learning, the relation labeling will become complex when the image is divided into 3 × 4 or more patches. Thus, we discard the auxiliary tasks to avoid this problem. Besides, we also discard the gap trick here to avoid generating tiny patches. In temporal structure learning, other experimental settings keep unchanged, and we evaluate the performance on UCF-101. As shown in Table 6, the performance does not always increase as the number of patches increase. These results show that it is more difficult to solve the sequential recovery task when the sequence size gets larger, and the information redundancy among samples also increases.

Table 7
The influence of image size.

Image size	UCF-101 (%)
64 × 64	57.3
80 × 80	59.6
100 × 100	60.5
120 × 120	59.3

Table 8
The effect of feature fusion module.

	STL-10 (%)	UCF-101 (%)	HMDB-51 (%)
Original method	77.76 ± 0.19	58.8	25.1
No fusion module	41.55 ± 0.22	35.1	12.7

Image size. The size of each image in the sequence also has an influence on the learned feature. When applying our method for spatial structure learning, the image size is related to the sequence size. To avoid it, we only use the temporal structure learning to evaluate the influence of image size. At the pre-processing step, we randomly cut a local region out of each selected frame to produce the training sequence. We vary the size of the cutting region at the pre-processing step as shown in Table 7. The performance increases as the image size increase, but it does not give much improvement beyond a size of 80 × 80. Due to the structure of fully connected layers, the image size significantly affects the number of parameters and the training time. Thus we prefer the small size.

The feature fusion module. The feature fusion module is capable of generating global features for the whole sequence. The network needs both global and local information to predict the index of the corresponding image in the sequence. To prove this argument, we remove the fusion module and train the model to solve the principal tasks. We compare the performance in Table 8. Apparently, removing the fusion module leads to a learning collapse.

5. Conclusion

In this paper, we propose a multi-task learning framework for the sequence recovery task, which is capable of handling both spatial sequence and temporal sequence. Compared to closely related works, our method is flexible and scalable to deal with sequences of an arbitrary length, and exploit the whole space of permutations. Besides, it is also generic to solve any ranking problem, like relative attributes. In the experiments, we prove that the proposed method learns competitive representation for semantic-level tasks. It outperforms the state-of-the-art results for unsupervised representation learning on STL-10 dataset, and it achieves competitive performance for action recognition as a pretraining method on UCF-101 and HMDB-51.

CRediT authorship contribution statement

Qilu Zhao: Conceptualization, Methodology, Software, Validation, Investigation, Writing - original draft. **Junyu Dong:** Resources, Writing - review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Thanks for all the supports we have. This work was supported by the National Natural Science Foundation of China [grant numbers 41576011, U1706218] and the Natural Science Foundation of Shandong Province of China (ZR2018ZB0852).

References

[1] T. Chen, S. Kornblith, M. Norouzi, G.E. Hinton, A simple framework for contrastive learning of visual representations, 2020, CoRR abs/2002.05709.
[2] A. van den Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, 2018, CoRR abs/1807.03748.
[3] A. Kolesnikov, X. Zhai, L. Beyer, Revisiting self-supervised visual representation learning, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, Computer Vision Foundation / IEEE, 2019, pp. 1920–1929.
[4] A. Dosovitskiy, P. Fischer, J.T. Springenberg, M.A. Riedmiller, T. Brox, Discriminative unsupervised feature learning with exemplar convolutional neural networks, IEEE Trans. Pattern Anal. Mach. Intell. 38 (9) (2016) 1734–1747.
[5] M. Noroozi, P. Favaro, Unsupervised learning of visual representations by solving jigsaw puzzles, in: Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI, 2016, pp. 69–84.
[6] R.S. Cruz, B. Fernando, A. Cherian, S. Gould, Deeppermnet: Visual permutation learning, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, 2017, pp. 6044–6052.
[7] G. Larsson, M. Maire, G. Shakhnarovich, Learning representations for automatic colorization, in: Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV, 2016, pp. 577–593.
[8] K. He, H. Fan, Y. Wu, S. Xie, R.B. Girshick, Momentum contrast for unsupervised visual representation learning, 2019, CoRR abs/1911.05722.
[9] S. Gidaris, P. Singh, N. Komodakis, Unsupervised representation learning by predicting image rotations, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, 2018.
[10] Y. LeCun, B.E. Boser, J.S. Denker, D. Henderson, R.E. Howard, W.E. Hubbard, L.D. Jackel, Handwritten digit recognition with a back-propagation network, in: Advances in Neural Information Processing Systems 2, [NIPS Conference, Denver, Colorado, USA, November 27-30, 1989], 1989, pp. 396–404.
[11] M. Raghu, C. Zhang, J.M. Kleinberg, S. Bengio, Transfusion: Understanding transfer learning for medical imaging, in: H.M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E.B. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, 2019, pp. 3342–3352.
[12] H. Lee, A. Battle, R. Raina, A.Y. Ng, Efficient sparse coding algorithms, in: Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006, 2006, pp. 801–808.
[13] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications, Neural Netw. 13 (4–5) (2000) 411–430.
[14] H. Bourlard, Y. Kamp, Auto-association by multilayer perceptrons and singular value decomposition, Biol. Cybernet. 59 (4) (1988) 291–294.
[15] N. Srebro, J.D.M. Rennie, T.S. Jaakkola, Maximum-margin matrix factorization, in: Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada], 2004, pp. 1329–1336.
[16] J.A. Hartigan, M.A. Wong, Algorithm AS 136: A k-means clustering algorithm, J. R. Stat. Soc. 28 (1) (1979) 100–108.
[17] X. Liu, X. Zhu, M. Li, L. Wang, E. Zhu, T. Liu, M. Kloft, D. Shen, J. Yin, W. Gao, Multiple kernel k-means with incomplete kernels, IEEE Trans. Pattern Anal. Mach. Intell. 42 (5) (2020) 1191–1204.
[18] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, Time-contrastive networks: Self-supervised learning from video, in: 2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018, 2018, pp. 1134–1141.
[19] M.A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, F. Li, A. Garg, J. Bohg, Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks, in: International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019, 2019, pp. 8943–8950.

[20] Y. Zhang, Y. Yang, T. Li, H. Fujita, A multitask multiview clustering algorithm in heterogeneous situations based on LLE and LE, Knowl.-Based Syst. 163 (2019) 776–786.
[21] P. Agrawal, J. Carreira, J. Malik, Learning to see by moving, in: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015, 2015, pp. 37–45.
[22] A. Owens, J. Wu, J.H. McDermott, W.T. Freeman, A. Torralba, Ambient sound provides supervision for visual learning, in: Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, the Netherlands, October 11-14, 2016, Proceedings, Part I, 2016, pp. 801–816.
[23] L. Pinto, D. Gandhi, Y. Han, Y. Park, A. Gupta, The curious robot: Learning visual representations via physical interactions, in: Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II, 2016, pp. 3–18.
[24] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, A.A. Efros, Context encoders: Feature learning by inpainting, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, 2016, pp. 2536–2544.
[25] R. Zhang, P. Isola, A.A. Efros, Colorful image colorization, in: Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III, 2016, pp. 649–666.
[26] R. Zhang, P. Isola, A.A. Efros, Split-brain autoencoders: Unsupervised learning by cross-channel prediction, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, 2017, pp. 645–654.
[27] D. Pathak, R.B. Girshick, P. Dollár, T. Darrell, B. Hariharan, Learning features by watching objects move, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, 2017, pp. 6024–6033.
[28] M. Caron, P. Bojanowski, A. Joulin, M. Douze, Deep clustering for unsupervised learning of visual features, in: Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIV, 2018, pp. 139–156.
[29] C. Doersch, A. Gupta, A.A. Efros, Unsupervised visual representation learning by context prediction, in: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015, 2015, pp. 1422–1430.
[30] X. Yu, X. Ye, Q. Gao, Infrared handprint image restoration algorithm based on apoptotic mechanism, IEEE Access 8 (2020) 47334–47343.
[31] T.N. Mundhenk, D. Ho, B.Y. Chen, Improvements to context based self-supervised learning, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, 2018, pp. 9339–9348.
[32] M. Noroozi, A. Vinjimoor, P. Favaro, H. Pirsiavash, Boosting self-supervised learning via knowledge transfer, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, 2018, pp. 9359–9367.
[33] U. Ahsan, R. Madhok, I.A. Essa, Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition, in: IEEE Winter Conference on Applications of Computer Vision, WACV 2019, Waikoloa Village, HI, USA, January 7-11, 2019, 2019, pp. 179–189.
[34] H. Lee, J. Huang, M. Singh, M. Yang, Unsupervised representation learning by sorting sequences, in: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, 2017, pp. 667–676.
[35] I. Misra, C.L. Zitnick, M. Hebert, Shuffle and learn: Unsupervised learning using temporal order verification, in: Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I, 2016, pp. 527–544.
[36] B. Fernando, H. Bilen, E. Gavves, S. Gould, Self-supervised video representation learning with odd-one-out networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, 2017, pp. 5729–5738.
[37] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012, Proceedings of a Meeting Held December 3-6, 2012, Lake Tahoe, Nevada, United States, 2012, pp. 1106–1114.
[38] P.A. Viola, M.J. Jones, Rapid object detection using a boosted cascade of simple features, in: 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), with CD-ROM, 8-14 December 2001, Kauai, HI, USA, 2001, pp. 511–518.
[39] A. Coates, A.Y. Ng, H. Lee, An analysis of single-layer networks in unsupervised feature learning, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011, 2011, pp. 215–223.
[40] K. Soomro, A.R. Zamir, M. Shah, UCF101: A dataset of 101 human actions classes from videos in the wild, 2012, CoRR abs/1212.0402.
[41] H. Kuehne, H. Jhuang, E. Garrote, T.A. Poggio, T. Serre, HMDB: A large video database for human motion recognition, in: IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011, 2011, pp. 2556–2563.
[42] K.R. Jr., T. Tesafaye, MORPH: A longitudinal image database of normal adult age-progression, in: Seventh IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2006), 10-12 April 2006, Southampton, UK, 2006, pp. 341–345.
[43] J.J. Zhao, M. Mathieu, R. Goroshin, Y. LeCun, Stacked what-where auto-encoders, 2015, CoRR abs/1506.02351.
[44] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, in: Y. Bengio, Y. LeCun (Eds.), 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014.
[45] X. Ji, A. Vedaldi, J.F. Henriques, Invariant information clustering for unsupervised image classification and segmentation, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, IEEE, 2019, pp. 9864–9873.
[46] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, Beta-VAE: Learning basic visual concepts with a constrained variational framework, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017.
[47] A. Makhzani, J. Shlens, N. Jaitly, I.J. Goodfellow, Adversarial autoencoders, 2015, CoRR abs/1511.05644.
[48] J. Donahue, P. Krähenbühl, T. Darrell, Adversarial feature learning, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017.
[49] P. Bojanowski, A. Joulin, Unsupervised learning by predicting noise, in: D. Precup, Y.W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, in: Proceedings of Machine Learning Research, PMLR, vol. 70, 2017, pp. 517–526.
[50] R.D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, Y. Bengio, Learning deep representations by mutual information estimation and maximization, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net, 2019.
[51] P. Häusser, J. Plapp, V. Golkov, E. Aljalbout, D. Cremers, Associative deep clustering: Training a classification network with no labels, in: T. Brox, A. Bruhn, M. Fritz (Eds.), Pattern Recognition - 40th German Conference, GCPR 2018, Stuttgart, Germany, October 9-12, 2018, Proceedings, in: Lecture Notes in Computer Science, vol. 11269, Springer, 2018, pp. 18–32.
[52] A. Dundar, J. Jin, E. Culurciello, Convolutional clustering for unsupervised learning, 2015, CoRR abs/1511.06241.
[53] S. Yang, P. Luo, C.C. Loy, K.W. Shum, X. Tang, Deep representation learning with target coding, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA, 2015, pp. 3848–3854.
[54] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA, 2006, pp. 1735–1742.
[55] C. Vondrick, H. Pirsiavash, A. Torralba, Generating videos with scene dynamics, in: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, 2016, pp. 613–621.
[56] X. Wang, A. Gupta, Unsupervised learning of visual representations using videos, in: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015, 2015, pp. 2794–2802.
[57] M. Abadi, P. Barham, et al., Tensorflow: A system for large-scale machine learning, in: 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, Savannah, GA, USA, November 2-4, 2016, 2016, pp. 265–283.
[58] V. Nair, G.E. Hinton, Rectified linear units improve restricted Boltzmann machines, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel, 2010, pp. 807–814.
[59] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, 2015, pp. 448–456.
[60] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, CoRR abs/1412.6980.
[61] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in: Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009, 2009, pp. 41–48.