This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2022.3149329, IEEE Transactions on Circuits and Systems for Video Technology

Order-Constrained Representation Learning for Instructional Video Prediction

Muheng Li, Lei Chen, Jiwen Lu, Jianjiang Feng, and Jie Zhou

***Abstract*—In this paper, we propose a weakly-supervised approach called Order-Constrained Representation Learning (OCRL) to predict future actions from instructional videos by observing incomplete steps of actions. Most conventional methods focus on predicting actions based on partially observed video frames, which mainly study low-level semantics such as motion consistency. Unlike performing a single action, completing a task in an instructional video usually requires several steps of action and longer periods. Motivated by the fact that the order of action steps is key to learning task semantics, we develop a new frame of contrastive loss, called StepNCE, to integrate the shared semantic information between step order and task semantics under the framework of the memory bank-based momentum-updating algorithm. Specifically, we learn the video representations from step order-rearranged trimmed video clips based on the proposed task-consistency rule and order-consistency rule. Our StepNCE loss can be used to pre-train a video feature encoder, which is then fine-tuned to carry out the instructional video prediction task. Our approach digs deeper into the sequential logic between different action steps with respect to a certain task, which is able to promote the video understanding methods to a new semantic level. We evaluate our method on five popular instructional video and action prediction datasets: COIN, CrossTask, UT-Interaction, BIT-Interaction, and ActivityNet v1.2, and the results show that our approach gains improvements from conventional prediction methods.**

***Index Terms*—Instructional video, video prediction, weakly-supervised learning, representation learning.**

I. Introduction

Instructional videos have gained lots of attention on video analysis in recent years [1]–[4]. Differing from conventional action classification datasets [5], [6], instructional videos usually contain longer time scales and more complicated content. Vision-based prediction task is a conventional but meaningful field with various kinds of applications [7], which is also hard due to the incompleteness of information compared to fully-observed recognition tasks. Predicting the to-be-done tasks from incomplete action steps in instructional videos is even more challenging on account of the semantic complicacy among steps and between steps and the corresponding task. A key to understanding the connections between steps and the corresponding task is to study the intricate order constraints among steps (Fig. 1). It is tricky since the inversion of step order may still be semantically tenable without knowing what task is being performed in the video. For example, when performing a *glass vanishing* magic trick, the reversed step order would lead to a contrary show theme. To be specific, if the performer puts the curtain on an empty table, and opens it with a glass inside, the magic trick would turn out to be the *appearing glass* theme.

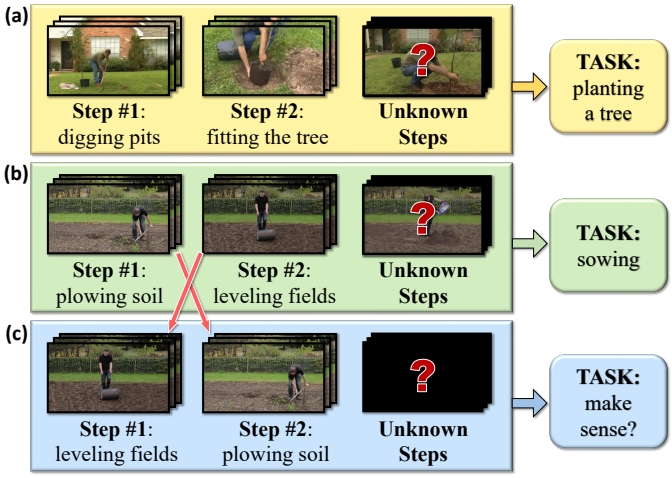

Fig. 1: **The complexity of instructional video task prediction.** Conventional action prediction focuses on predicting the action labels from partially-observed frame-level information. Predicting actions is less difficult than predicting tasks from instructional videos, since the latter makes predictions on a higher semantic level. It is hard to differentiate between the two tasks in (a) and (b) without exploiting the underlying semantic differences between *planting a tree* and *sowing*. In (c), the inversion of step order can be semantically meaningful, but leads to a diverse task.

The level of semantic abstraction is a pivotal nature of a video clip. For videos in action recognition datasets, the highest semantic abstraction is the action itself, while for instructional videos is the task. Usually, a task can include several semantic steps [8], of which each step can be regarded as a complete action. The conventional approach for early video prediction focuses on action-level prediction [9]–[12], which utilizes the lower-semantic-level information of partially observed time frames. The target semantic level of the prediction algorithm decides its scope of application. When dealing with video task prediction of incompletely observed instructional videos with a few pieces of steps, we have to exploit the complex relationship between different actions, and make final predictions based on a higher-level understanding

The authors are with the Beijing National Research Center for Information Science and Technology (BNRist), and the Department of Automation, Tsinghua University, Beijing 100084, China. E-mail: li-mh20@mails.tsinghua.edu.cn; chenlei2020@mail.tsinghua.edu.cn; lujiwen@tsinghua.edu.cn; jfeng@tsinghua.edu.cn; jzhou@tsinghua.edu.cn. *(Corresponding author: Jiwen Lu)*

The first two authors contribute equally.

of semantics. Furthermore, a major cause for why it is difficult to understand the interrelationship between tasks and steps is the sequentiality of step actions, since step order provides a strong reminder of what the final task is. Several temporal order-related approaches have been proposed for representation learning of videos [13]–[16]. Previous methods mainly focus on short-range temporal reasoning, which put more emphasis on semantically lower-level features including object displacements, continuities of actions or physical principles, *etc*. Time irreversibility is a shared characteristic concerning these low-level features. Conventional temporal-related approaches try to learn the physics of continuity. However, the temporal sequentiality of action steps in instructional videos is not based on physical information, but on the semantic understanding of the step-related task. When the order of steps is adjusted in an instructional video, it is not intuitive to find out the adjusted one without knowing the information of the target task. As a consequence, we have to dig deeper into the comprehension of complex semantics between step order and task, and are thus able to perform higher-level instructional video task predictions.

To address this problem, we propose the architecture, Order-Constrained Representation Learning (OCRL), for predicting instructional video tasks by exploring the relationship between step order and the semantics of the targeted task. We promote the conventional action prediction objective to a higher-level video task prediction objective. Our approach is designed under a pre-training and fine-tuning strategy. To make effective video task predictions, we first try to exploit the complicated semantic relationship between step order and task by introducing a new StepNCE contrastive loss. In detail, We extract video representations from step-accordingly trimmed video clips. The video representations are extracted by a pre-trained contrastive model utilizing an order-constrained discipline, which defines the positive samples under the rules of order consistency (OC) and task consistency (TC). The pre-trained model is then fine-tuned to solve the problem of video task prediction. The main contributions of this paper can be summarized as three points:

1) We propose a weakly supervised approach to make video task predictions on instructional videos, which promotes the video prediction methods to higher-level semantical applications. Unlike previous early action prediction approaches, our objective is to predict certain tasks in instructional videos by observing part of the action steps rather than predict certain actions by observing part of unfinished movements. To the best of our knowledge, we are the first to make attempts on the issue of instructional video task prediction.
2) A new contrastive loss, StepNCE, is developed to learn the video representations, exploiting the high-level semantics between step order and task. We study the inner semantic consistencies of instructional videos, and introduce two sets of rules, *task consistency (TC)* and *order consistency (OC)*, to define the positive samples during contrastive learning for a better understanding of the relationship between tasks and steps. We use the StepNCE loss to extract features from trimmed target video clips for further application on task predicting objective.
3) We validate the efficiency of our method on several prediction-related datasets: COIN, CrossTask, UT-Interaction, BIT-Interaction, and ActivityNet v1.2. Ablation studies are conducted to study the choices of different parameters, and to prove the validity of our settings.

II. Related Work

In this section, we discuss some topics related to instructional videos, the progress in action prediction, and the research trends in self-supervised video representation learning.

A. Instructional Video

Learning from instructional video is an increasingly popular trend in video analysis. Numerous instructional video datasets have been proposed in recent years [4], [8], [17]. With the help of the collected datasets, various tasks have been studied around instructional videos. A principle trend is learning the representations from instructional videos. The current research direction of instructional representation learning can mainly be divided into two categories: joint vision and narration representation learning and vision-based representation learning. A series of work pays attention to learning the joint script and video representations [1], [3], [17], [18]. These works try to exploit the relationship between actions and narrations, but are inapplicable to deal with the data with only visual contents. Zhukov *et al.* [4] find a way to learn the cross-domain semantics between the action steps of different tasks. Kukleva *et al.* [19] learn the temporal embedding based on continuous frames, which exploits the specific order certain actions are performed. Xu *et al.* [20] introduce a boundary-sensitive pretext (BSP) task to learn video representation. The above methods mainly concentrate on the video itself, and delving information from the visual features. There are also some works trying to discover the usefulness of step order. Similar to our work, Zhukov *et al.* [21] utilize the information of long-range step order to estimate the actionness of each segment. They mainly use the representations to perform the downstream task of action localization. The pivotal innovation of our model is that we try to discover the relationship between semantic task and step order across various domains using the contrastive method, and perform a novel task prediction objective.

B. Vision-based Prediction

Making predictions based on visual contents serves as the fundamental approach in applications like robotics, surveillance, scene understanding, *etc*. Visual prediction can be divided into several sub-tasks with different kinds of inputs and outputs according to [7]. For example, video prediction tries to predict future scenes through observed video frames [22]–[26], video-based emotion prediction aims at analyzing and predicting human emotions from the information in video clips [27]–[29], and action anticipation concentrates on predicting the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2022.3149329, IEEE Transactions on Circuits and Systems for Video Technology

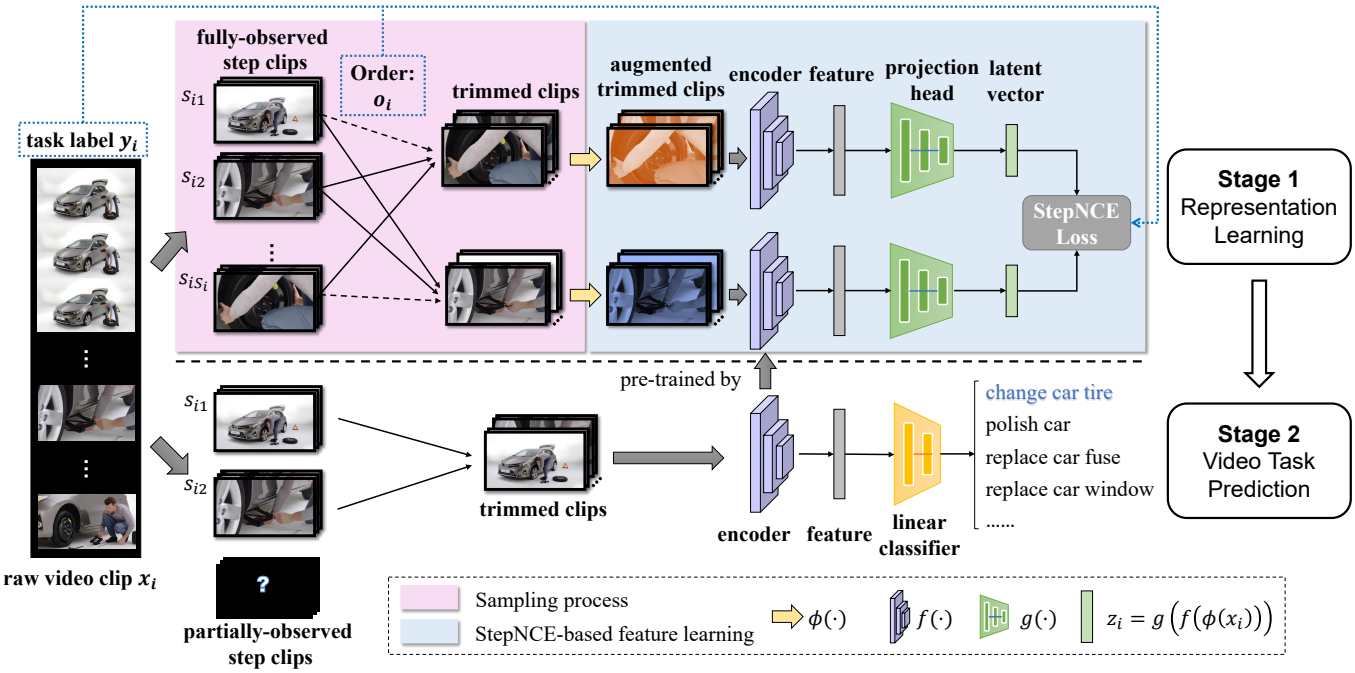

Fig. 2: **Framework of our proposed OCRL method.** OCRL is mainly composed of two parts. The above part refers to the representation learning architecture, and the lower part refers to the video prediction architecture.

next step of action or event [30]. Our work should fall under the category of early video prediction, which predicts the main video content through peeking at an incomplete component of the video. In the field of early video prediction, Kong *et al.* [9] *et al.* propose the Adversarial Action Prediction Networks (AAPNet) to predict the action label from a partially observed video. Aliakbarian *et al.* [31] develop a two-stage LSTM architecture and a new loss to predict the action. Chen *et al.* [32] extract features from human body parts and uses the attention module to perform action prediction. Wang *et al.* [10] utilize a teacher-student architecture to distill progressive knowledge and make predictions. Wu *et al.* [33] predict actions by reasoning about the spatial-temporal relations between persons and contextual objects. Liu *et al.* [34] develop an observation ratio regression module to learn stronger representations similar to full video representations and discriminative for action prediction. The existing approaches for early video prediction mostly cluster on early action prediction, which predicts the action label using partial video frames. Our objective is to promote the prediction task to a higher semantic level. Instead of anticipating the actions, we predict the semantic task labels of instructional videos using partially observed action steps.

C. Self-supervised Representation Learning in Videos

Since the prosperity of self-supervised representation learning in the image domain, a succession of studies focusing on the video domain has been conducted as well. Several pretext tasks are designed to learn video representations without supervision such as predicting the future [35], [36], predicting playback rate [37], solving space-time cubic puzzles [38] and predicting motion and appearance statistics [39]. Some previous works are similar to our approach including predicting the arrow of time [14] and predicting the shuffled frames [40] and video clips [15], [41]. The previous approaches mainly focus on local time relations, while the attention on the global temporal relationships in long videos (*e.g.* instructional videos) is the main concern of our work.

In addition, the contrastive learning framework [42], [43] has been a great success for self-supervised learning. Diverse ways of defining positive and negative samples are designed to learn representations from videos. For example, DPC [44] and MemDPC [45] regard the predicted and ground-truth feature of a certain clip at the same spot as positive samples, CoCLR [46] utilizes both RGB and optical flow information to define positive samples and trains the model coherently, and Pace [47] uses the videos of the same class but different playback speed as positive samples. SeCo [48] uses diverse frames from the same input videos as positive samples, while CVRL [49] utilizes diverse clips from the same video. Moreover, there is a trend of video-related contrastive learning approaches which focus on cross-modality learning. For instance, audio information [50], [51], or narrations [3] can be integrated together with videos to generate contrastive loss. Our work learns the core idea of contrastive learning strategy, and defines the positive and negative samples under two proposed rules, task consistency (TC) and order consistency (OC), which will be covered in the following section.

III. Approach

In this section, we will first introduce the newly proposed Noise Contrastive Estimation (NCE) loss, StepNCE, which combines the information of semantic step order and task label. Then, we illustrate our memory bank-based momentum-updating algorithm and network architectures in detail.

TABLE I: **Referring for all important notations**

Notation	Referring
$\mathcal{X}$, x	set of video clips, video clip
$\mathcal{X}'$, x'	set of augmented video clips, augmented video clip
$\mathcal{L}$, y	set of task labels, task label
$\mathcal{O}$, o	set of order labels, order label
$\mathcal{S}_{x_i}$, s_{ij}	set of steps for video clip x_i, step j for x_i
S_i	count of total steps of x_i
S_p	count of observed steps
l, l_s	frame length of full video/step segment
z_i	latent vector of x_i for NCE loss
z_p, z_n	positive/negative latent vector pair of z_i
$\mathcal{P}_i$, $\mathcal{N}_i$	positive/negative set of data for input x_i

In consideration of readability, all the important notations are summarized in Table I. Let $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ denote N raw video clips. The corresponding task label set is defined as $\mathcal{L} = \{y_1, y_2, \ldots, y_N\}$. S_i different steps are contained in each video clip x_i in total, which can be denoted as $\mathcal{S}_{x_i} = \{s_{i1}, s_{i2}, \ldots, s_{iS_i}\}$. For the kth step s_{ik} in x_i, a start frame and an end frame are annotated to conduct a segmented video clip for the step. The goal of instructional video task prediction is to predict the task label y_i of x_i by watching only the beginning S_p steps out of S_i, that is to say, we concatenate the first S_p clips of the successive steps in each raw video x_i to generate a to-be-predicted trimmed video set.

The purpose of trimming is to extract useful information and carry out efficient video task prediction. Instructional videos usually contain redundant frames with useless visual information, *e.g.* narrating or teasing, according to [21]. The semantically uninformative contents are likely to be randomly distributed, which makes them less sequentially correlated and provides no contribution for predicting the task labels. Thus, we trim the long videos by step annotations, and concatenate the trimmed video segments of steps to produce the input samples.

The pipeline of our method is shown in Fig. 2. OCRL is mainly composed of two parts: the representation learning part and the video prediction part. Raw video clips are first inputted into the representation learning architecture to train a semantically discriminating encoder based on weakly supervised StepNCE loss. The learned encoder is then used as a feature extractor before the classifier in the video prediction architecture. We will cover the details of both architectures in the following sub-sections.

A. Representation Learning: StepNCE

To learn the representations of input video clips, we design a weakly supervised StepNCE-based feature extractor for learning the representations of input video clips.

Consistency Analysis. The semantics of instructional video task not only depends on the components of its consisting actions, but also on the order of the action steps. An ordered concatenation of 3 action steps, $\mathcal{S} = \{s_1, s_2, s_3\}$, is combined to form a semantically valid task T. We could use the *glass vanishing* magic trick example for detailed explanation, then s_1, s_2, s_3 would respectively be *displaying the glass*, *putting on the curtain*, *displaying the empty table*. When we introduce another semantic action s_{new}, for example *breaking the glass*, and replace it with one of the composing actions, producing $\mathcal{S} = \{s_1, s_{new}, s_3\}$, it may lead to a different semantics of task, or semantic confusions caused by uncorrelated actions (e.g. we use *removing the tyre* as s_{new}). We name this phenomenon as *task consistency (TC)*. In addition, when we reverse the step order, producing $\mathcal{S} = \{s_3, s_2, s_1\}$, it will also lead to an temporally-inverse semantics (*i.e. appearing glass trick*) or cause sequential invalidation if the semantics between steps are causal and irreversible. We name this phenomenon as *order consistency (OC)*.

The goal of consistency analysis is to analyze the design mechanism of our method. The objective is to learn the internal logic between different action steps, as well as between steps and tasks, by generating positive/negative data pairs for the model to compare and discriminate. The TC-rule and OC-rule should be considered and preserved. The model learned from the two consistency rules could provide plenty of useful information for the instructional task predictions.

Sampling. Sampling procedure of the training data for representation learning is slightly different from the data sampling procedure of final video task prediction. We choose to consider S_p steps to form input clips. Rather than extracting the first S_p steps, we randomly select S_p steps out of S_i to make full use of the step-annotated data. To integrate information of step order into the input samples, one possible approach is to concatenate the clips in a random permutation of order with the chosen S_p steps. However, this is not an applicable way to generate easy-to-learn signals, since S_p steps can generate $S_p!$ types of permutations in total, and the number will become excessively large with the increase of S_p. Furthermore, the semantics of temporally long-range step permutations tend to become over-complicated. A little adjustment of the interior step order could not affect the exterior semantics of the task label in certain tasks, *e.g.* a person can either choose to prepare the filler or cut the bread first when he/she is making a sandwich. Due to the reasons above, we incorporate the information of step order into our pre-training approach by randomly reversing the step order to generate temporally positive and negative sequences.

Our architecture utilizes a fixed time length for input clips. Suppose the input video format of our network is $c \times l \times h \times w$, where c refers to channel number, l refers to frame number, h and w are the height and width of each frame. If we select S_p steps for observation, then the clip length for each step would be $l_s = l/S_p$ frames. Here, we randomly crop the clip through time dimension in a step if its length is longer than l_s, or pad the clip using the end frame of step if its length is shorter than l_s.

Disorders can occur due to the editing process in instructional videos despite the sequentiality of humans performing tasks. For instance, the film editor can display the visual contents of the final target at the beginning of a video, then explain how to achieve the goal. In some datasets like COIN [8], there is a unified label system that gives all the tasks with their containing steps. All the videos are segmented and labeled under the pre-defined label system. Thus, we can address this problem by rearranging the action steps that happen in a video clip by the order of sequential semantics according to the prior knowledge from the label system. In addition, repeated step

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2022.3149329, IEEE Transactions on Circuits and Systems for Video Technology

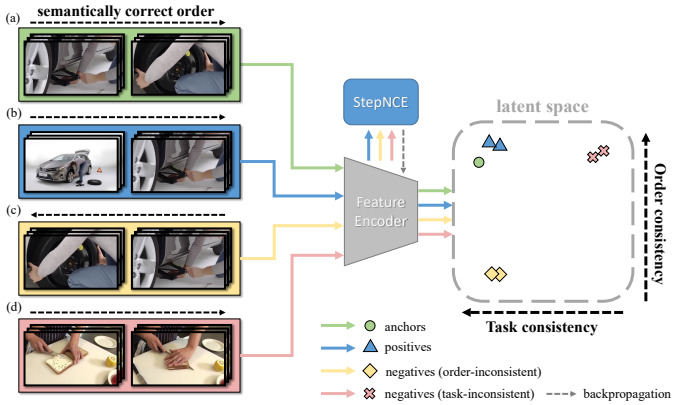

Fig. 3: **Illustration for the StepNCE-based contrastive representation learning process.** Samples (a)-(d) are trimmed from raw video clips under our sampling protocols. We define positive and negative samples with regard to *task consistency* and *order consistency*. In StepNCE, the similarity between anchors and positive encoded features are maximized, and vice versa. We acquire the optimal feature encoder via contrastive StepNCE learning.

combinations are eliminated through data pre-processing as well, since these combinations usually indicate that the same task executes repeatedly inside a single video.

StepNCE. Noise Contrastive Estimation (NCE) [52] has shown its reliable performances on self-supervised visual representation learning recently. NCE tries to distinguish target samples from noise samples, thus extracts the key information from the data. The instance discrimination loss, introduced by [42], learns the sample-based differences between different views of object, and extracts self-supervised signals for downstream tasks including image classification [43], action recognition [46], *etc*.

To begin with, we introduce the encoder mapping $f : \mathcal{X} \rightarrow \mathbb{R}^d$, which embeds an input video clip x into a d-dimensional feature vector $f(x) \in \mathbb{R}^d$. Suppose there is a video augmentation mapping $\phi : \mathcal{X} \rightarrow \mathcal{X}'$, which randomly applies a visual-augmenting transformation to video clip x, generating an augmented version x'. To create the instance-discriminated sample pairs for contrastive learning, we define the positive samples as all kinds of randomly augmented $\phi(x_i)$ for x_i, while the negative set contains $\mathcal{N}_i = \{\phi(x_n) \mid \forall n \neq i\}$. For convenience we define $z_i = f(\phi(x_i))$, then the instance discrimination loss can be expressed as:

$$\mathcal{L}_{ID} = -\mathbb{E}_{\mathcal{X}} \left(\log \frac{e^{z_i \cdot z_p/\tau}}{e^{z_i \cdot z_p/\tau} + \sum_{n \in \mathcal{N}_i} e^{z_i \cdot z_n/\tau}} \right) \quad (1)$$

where τ refers to the softmax temperature, and $\cdot$ refers to dot product operator. The exponential of dot product between vector z_i and z_p (z_n) measures the joint probability score of the embedded feature pair. From (1), the instance discrimination loss maximizes the joint probability score of positive feature pairs, which also minimizes that of negative feature pairs. In this way, the ideal encoder can be trained as an optimum model to discriminate the diverse instances from each other regardless of the noises caused by the visual augmentation functions.

To acquire more information through representations of instructional videos, a better design of contrastive loss is required. (1) only focuses on instance-level information, which lacks the consideration of higher semantic knowledge. (1) has the ability to discern the target instance from its visual-augmented version, but provides few information for understanding which semantic task it should belong to. To include more prediction-relevant signals into our loss function, we propose a weakly supervised version of contrastive loss, StepNCE, for instructional video prediction. According to the consistency analysis, task label and step order are both valuable information for the to be completed task. Thus, we will introduce both TC-rule and OC-rule into the StepNCE loss function. In Sampling part, we have randomly applied the reverse-order operation to the sampled steps. Suppose we denote the corresponding step order set of $\mathcal{X}$ as $\mathcal{O} = \{o_1, o_2, \ldots, o_N\}$, where o_i equals to 0 when the step order of x_i has been reversed, and to 1 vice versa. To integrate the information of task label and step order into the basic idea of the previous instance discrimination loss, we redefine the positive set and the negative set for x_i as

$$\mathcal{P}_i = \{\phi(x_p) \mid o_p = o_i \ and \ y_p = y_i, \ \forall p \in [1, N]\} \quad (2)$$

$$\mathcal{N}_i = \{\phi(x_n) \mid o_n \neq o_i \ or \ y_n \neq y_i, \ \forall n \in [1, N]\} \quad (3)$$

Then StepNCE loss can be expressed as

$$\mathcal{L}_S = -\mathbb{E}_{\mathcal{X}} \left(\log \frac{\sum_{\mathrm{p} \in \mathcal{P}_i} e^{z_i \cdot z_p/\tau}}{\sum_{\mathrm{p} \in \mathcal{P}_i} e^{z_i \cdot z_p/\tau} + \sum_{n \in \mathcal{N}_i} e^{z_i \cdot z_n/\tau}} \right) \quad (4)$$

Here, we preserve the TC-rule and OC-rule by matching the positive pairs of samples with regard to both action and sequence correspondence. The TC-rule refers to weakly supervised information, while the OC-rule refers to self-supervised information. The query feature can be positively matched with the key feature only when both the task and order are consistent with each other. We aim to increase the similarities between the embedded feature vectors of positive samples (otherwise irrelevance for negative samples) by minimizing the loss function in (4), thus enhancing the model to learn the inner semantics between observed actions and target tasks. Intuitively, we still take the task of *glass vanishing* magic trick as an example. The action steps (*displaying the glass→putting on the curtain→displaying the empty table*) are contributed to the task semantics both by their respective intra-action semantics and inter-action sequential logic. Our proposed StepNCE loss takes sample pairs as positive only when both the intra-action semantics and inter-action order are consistent with each other, which indicates that they share the same task semantics. By distinguishing samples from both the task semantics and step order at the same time, StepNCE gains the stronger capability to learn the task semantics from multiple sequential action steps.

Discussion: The way of defining positive/negative samples is the key to the contrastive learning framework. A good view for the contrastive representation learning of a specific task

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2022.3149329, IEEE Transactions on Circuits and Systems for Video Technology

should preserve as much task-related information as it can while discarding the uncorrelated information [53]. When we design the representation learning framework for instructional video prediction, the most important principle is: what kind of views gives the most effective hints for predicting the to-be-done task? The task-consistency rule is a very direct way that indicates the task information. According to our consistency analysis, the order-consistency rule indirectly implies the task information as well. However, using only the two consistency rules is insufficient for the pre-training stage, since the rules do not change the uncorrelated information between the two generated views. This is why we induce the random sampling strategy together with the visual augmentation functions that are applied on video clips. In addition, our model is pre-trained on an instance discrimination-based model, which means that the model has already obtained some knowledge to wipe out uncorrelated information.

Moreover, our proposed approach depends on three types of annotation information to learn from instructional video:

1) The semantic annotations for all action steps.
2) The positional annotations for all action steps.
3) The semantic annotations for the task of the video.

The aforementioned annotations are the required information for the objective of instructional video task prediction. The first two types of annotation information are needed for the sampling procedure, and the third type of annotation information is needed for both the representation learning part and the prediction part.

B. Algorithm Design

To improve the training effectiveness, we need to modify the contrastive learning strategy for better performance. The original contrastive learning framework is to compare the two diverse augmented versions of each sample, which induces the instance discrimination loss [43]. However, our StepNCE loss introduces new positive/negative pairs under TC-rule and OC-rule, which also requires more techniques to train an optimum encoder. Here, we optimize our algorithm by using memory banks and the momentum update strategy referring to the idea of MoCo [54].

In detail, we introduce a feature bank $queue_f$ together with two extra banks $queue_y$ and $queue_o$ to store the task labels and the order labels respectively. During each training iteration, the input data x_i, together with its task label y_i and order label o_i, will be separately used as query data and key data. In particular, the sample order of the input batch is shuffled, and then distributed to the GPUs for training. The encoded feature order is reshuffled back to the original order. The concurrently generated query and key data will always be a positive pair since they are augmented from the same trimmed input. Each time, the query input will compare with all the key data in memory banks, and compute the StepNCE loss. Following the momentum update strategy in [54], the parameters of the encoder f_k as θ_k is updated by:

$$\theta_k = m\theta_{k-1} + (1-m)\theta_q \tag{5}$$

where the parameters of the encoder f_q as θ_q is updated by back-propagation, and m refers to the momentum coefficient.

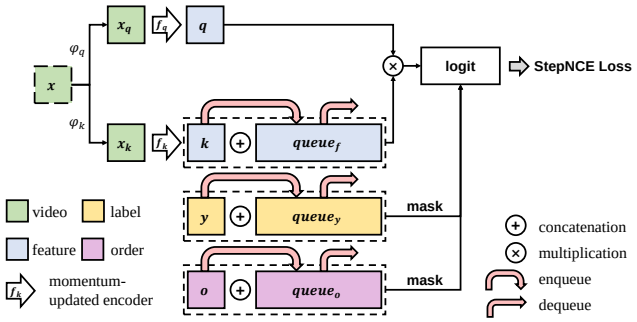

Fig. 4: **Illustration for the algorithm design of StepNCE loss updating process.** The sampled video clip x is inputted into the framework together with its corresponding task label y and order label o. Totally three memory banks are required for the updating process. The back-propagation process will update the parameters of encoder f_q, while the parameters of f_k are smoothly updated by the momentum mechanism.

The extracted feature of key k and its corresponding task label y_k and order label o_k is then stored in the memory banks for future updates. Differing from the original contrastive learning framework, memory banks are pivotal in our method since the comparing criteria are far more complex, which means multiple comparisons are necessary for our approach to learn something useful. The momentum update strategy is key to the usage of memory banks, since it makes the parameters of the key encoder grow more smoothly, and prevents the failure of the memory bank-based training process.

C. Network Architecture

Representation learning. To embed the trimmed video clips into feature vectors, we decide to use S3D [55] architecture as the backbone network, which plays the part of the encoder function $f(\cdot)$. In the pre-training stage, we add an MLP projection head $g(\cdot)$ with one hidden layer after the backbone network, which projects the 1024-dimensional backbone output to a 128-dimensional feature vector. By the reduction of dimensionality, the contrastive training framework can work more efficiently, and save more memory space. During the updates of our StepNCE loss, we adopt the MoCo-like strategy for training. For the visual augmentation function ϕ, we randomly apply size cropping, color distortions, gray-scaling, Gaussian blur, and horizontal flipping to input clips.

Instructional video prediction. After the video representation is extracted, the pre-trained encoder $f(\cdot)$ is then fine-tuned to perform the instructional video prediction task. The previous MLP projection head is abandoned in the prediction stage, and replaced by a fully connected classifier for prediction.

IV. Experiments

A. Datasets

We evaluated our method on the COIN dataset [8], the CrossTask dataset [56], the UT-Interaction #1, the UT-Interaction #2 [57], the BIT-Interaction dataset [58] and the ActivityNet v1.2 dataset [59].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2022.3149329, IEEE Transactions on Circuits and Systems for Video Technology

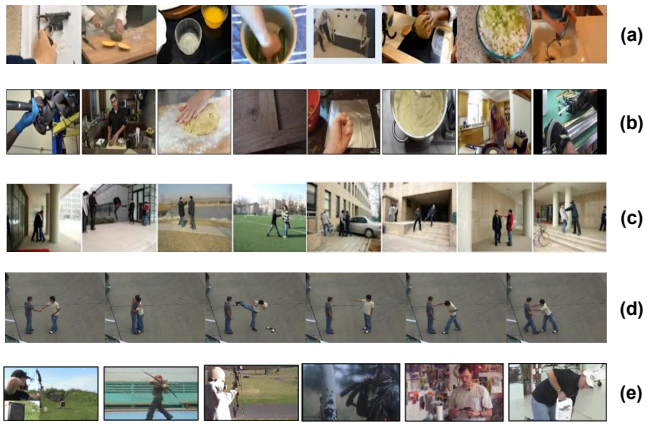

Fig. 5: **Example videos of five datasets.** (a) COIN dataset, (b) CrossTask dataset, (c) BIT dataset, (d) UTI dataset. (e) ActivityNet dataset.

COIN dataset. COIN dataset collects its samples from YouTube. It contains untrimmed instructional videos of 180 semantic tasks in 12 different domains (*e.g.*, sports, vehicles, *etc*.). COIN is organized in a three-level semantic architecture, of which each video is sequentially annotated into several steps of actions. The advantages of fine-grained annotations in COIN for action steps provide prerequisite information for our step-based task prediction. A total of 11,827 videos are contained in COIN.

CrossTask dataset. CrossTask dataset collects instructional videos of 83 tasks related to cooking, car maintenance, crafting, and home repairs from YouTube. Each task in CrossTask is divided into several action steps referring to wikiHow, a website that describes the task solving process. The original dataset is used to investigate the sharing information of interrelated tasks. We utilize the data of 18 primary tasks in CrossTask to validate our task prediction objective, and the remaining related tasks are discarded concerning the lack of labeled information of steps.

UTI #1 and UTI #2 datasets. The UT-Interaction dataset contains videos of continuous executions of 6 classes of human-human interactions: hand-shake, point, hug, push, kick and punch. Each video contains at least one execution per interaction. The videos are divided into two sets, of which each set contains 60 videos. The prediction of UTI Datasets is quite valuable since the UTI Datasets compare acts of violence to violence-like actions. The prediction results are useful for alerting before the occurrence of criminal behaviors.

BIT-Interaction dataset. The BIT-Interaction dataset contains a set of 8 classes of different human actions with 400 realistic videos. People in each interaction class produce different behaviors. The eight equal-sized classes (50 per action) of human interactions (bow, boxing, handshake, high-five, hug, kick, pat, and push) are contained in BIT in total. Prediction on the BIT-Interaction has similar usages to the UT-Interaction dataset as preventing crimes.

ActivityNet v1.2 dataset. ActivityNet v1.2 is a popular large-scale human activity recognition dataset that consists of 4,819 training, 2,383 validation, and 2,480 test videos with 100 categories. Since the labels are not provided for the test set, we instead use the validation set for testing.

B. Implementation Details

We implemented our OCRL model on the Pytorch toolbox under Python 3.8. We conducted experiments on a server with the Intel(R) Xeon(R) E5-2620 v4 CPU @ 2.10Ghz with 256GB RAM, and deep networks were trained on four NVIDIA GTX 1080 Ti GPUs with 11GB VRAM.

Representation learning. We pre-train our backbone network on the COIN dataset and the CrossTask dataset. For the tested data, we down-sampled the videos at 10 fps, with a spatial resolution of 128×128 pixels. The input video length is set to be 32 frames, which covers about 3 seconds in total. For certain S_p values, the step clip length can not be evenly divided (for example $S_p = 3$), we just lengthen the sampled clip of the last step to fill in 32 frames. Thus, the input size of the pre-trained network ($c\times l\times h\times w$) is set to be $3\times32\times128\times128$.

The data augmentation approaches include **size cropping**, **color distortion**, **gray-scaling**, **Gaussian blur**, and **horizontal flipping** to input clips. During the **size cropping** operation, the original input size, which is 224×224, is randomly cropped and resized to 128×128 instead. The cropped frame area is randomly chosen between $(0.2, 1.0)$ of the original frame area, and the cropped aspect ratio is chosen between $(3/4, 4/3)$. The limitations would avoid invalid cropping results. During the **color distortion** operation, the brightness, contrast, and saturation attributes of the input frames are randomly adjusted to $(0.6, 1.4)$-fold. The hue is randomly jittered between $(-0.1, 0.1)$-fold. The color distortion operation is randomly processed with an 80% possibility. The **gray-scaling** operation randomly generates gray-scale clips with a 20% possibility. The **Gaussian blur** operation randomly blurs the input frames with a Gaussian kernel with $\sigma \in [0.1, 2.0]$. The Gaussian blur operation is randomly processed with a 50% possibility. The **horizontal flipping** operation randomly flipped the input frames horizontally with a 50% possibility. All the augmentation methods are finished clip-wise, which means that each frame in the same video clip uses the same set of augmentation parameters. Furthermore, we have defined two transformation sets: the base transformation set and the full transformation set. The former one only contains the **size cropping** and **horizontal flipping** operations, while the latter one contains all of the above transformations. Beyond the two transformation sets, we have another two augmentation strategies while generating query and key data views: by randomly augmenting the two clips with diverse sets, or augmenting each clip with a randomly selected set. The two strategies are randomly selected with an equal possibility, which results in affluent patterns of augmented views.

During the pre-training stage, we use the ADAM [60] optimizer. The initial learning rate is 10^{-3} with a 10^{-5} weight decay. When the validation loss plateaus, the learning rate scales down by 0.1. For the momentum updating strategy, we use softmax temperature $\tau = 0.07$, momentum $m = 0.999$, with a queue size of 16384. The batch size for model pre-training is 32 samples for each GPU. Our model is trained

TABLE II: **The prediction accuracy (%) on COIN when $S_p = 3$.** Tasks with a different number of steps are evaluated. OCRL(i) and OCRL(ii) refer to two training strategies for the classifier respectively. To be noticed, the size of the dataset for each step number is not evenly distributed. (Comparing methods are all trained under fine-tuning strategies; *ss* and *ws* are respectively refer to self-supervised method and weakly supervised method.)

Methods ($S_p = 3$)	**Number of steps (size of test set)**			
	4 (518)	5 (125)	6 (26)	7 (8)
S3D (scratch) [55]	14.1	12.1	38.5	0
InfoNCE-based(*ss*) [42]	25.7	30.6	46.2	12.5
UberNCE-based(*ws*) [46]	42.1	47.6	61.5	**50.0**
OCRL(i)(*ws*) (ours)	51.4	58.9	**69.2**	37.5
OCRL(ii)(*ws*) (ours)	**54.4**	**62.9**	61.5	**50.0**

TABLE III: **The prediction accuracy (%) on COIN with diverse choices of S_p.** The results show that the performances of different S_p values could be related to the total step counts.

Method (S_p)	**5 steps**			**6 steps**		
	2	3	4	2	3	4
InfoNCE-based(*ss*) [42]	32.2	30.6	28.2	42.3	46.2	57.7
UberNCE-based(*ws*) [46]	50.8	47.6	50.0	65.4	61.5	57.7
OCRL(i)(*ws*) (ours)	48.4	58.9	46.0	57.7	**69.2**	53.9
OCRL(ii)(*ws*) (ours)	47.6	**62.9**	54.8	50.0	61.5	**69.2**

on 2 GPUs with 300 epochs. The base model on which we pre-trained our StepNCE model is trained by an instance discrimination loss on UCF101 [6] dataset for 400 epochs.

Instructional video prediction. During the prediction stage, we replace the MLP projection head with a fully connected classifier. We adopt two strategies for training the classifier: **(i)** directly train the classifier without updating the parameters in the backbone network; **(ii)** train the classifier with the backbone network fine-tuned. We use a learning rate of 10^{-3} with a weight decay of 10^{-3} for strategy **(i)**, while for strategy **(ii)** we choose the learning rate as 10^{-4} concerning the updates on backbone parameters. We adopt a cross-entropy loss and the ADAM optimizer for training. For the training inputs, we randomly select S_p steps out of S_i to form the trimmed input clip for x_i. We did not choose the first S_p steps to enhance the generalization prediction ability, since the actions of the first S_p steps for the same task can be nonidentical. However, we only preview the first S_p steps during validation.

We also use some data augmentation techniques during the training stage for the data generalization of the prediction model. The input data is **size cropped** and **color distorted** for the training stage. The size cropping parameters are the same as the ones in the pre-training stage. The color distorting parameters are also the same, except for the 30% processing possibility. We also apply the **size cropping** augmentation to input videos during the validation stage.

C. Performances on Instructional Video Prediction

We evaluate the prediction performances of OCRL on the instructional video dataset: COIN and CrossTask. We utilize the training/testing settings in [8] and [56] respectively. We compare our OCRL method with the train-from-scratch method [55] and several other contrastive-based methods [46] and [42]. To be noticed, the above methods have not included instructional video prediction as a tested case. We implemented those methods and tested them on the target datasets for comparison purposes.

TABLE IV: **The prediction accuracy (%) on CrossTask when $S_p = 4$.** Tasks with a different number of steps are evaluated. (Comparing methods are all trained under fine-tuning strategies.)

Methods ($S_p = 4$)	**Number of steps (size of test set)**			
	5 (70)	6 (53)	7 (33)	8 (31)
S3D (scratch) [55]	40.0	48.1	40.6	23.3
InfoNCE-based(*ss*) [42]	42.9	40.4	50.0	30.0
UberNCE-based(*ws*) [46]	75.7	78.8	53.1	53.3
OCRL(i)(*ws*) (ours)	70.0	80.8	43.8	60.0
OCRL(ii)(*ws*) (ours)	**84.3**	**86.5**	**59.4**	**66.7**

TABLE V: **The prediction accuracy (%) on CrossTask with diverse choices of S_p.** The results show that the performances of different S_p values could be related to the total step counts.

Method (S_p)	**5 steps**			**6 steps**		
	2	3	4	2	3	4
InfoNCE-based(*ss*) [42]	25.7	35.7	42.9	38.5	44.2	40.4
UberNCE-based(*ws*) [46]	57.1	70.0	75.7	61.5	76.9	78.8
OCRL(i)(*ws*) (ours)	54.3	67.1	70.0	50.0	59.6	80.8
OCRL(ii)(*ws*) (ours)	67.1	**85.7**	84.3	63.5	73.1	**86.5**

In COIN, all the 11,827 videos are split into 9030 training samples and 2797 testing samples. To be specific, we pre-train the S_p-step model using the videos with no less than S_p steps on training samples. The videos with just S_p steps do not accord with our setting for video prediction, but are used as pre-training samples since the pre-trained model mainly captures the relationship between step order and task. Those data are excluded during validation. Later, we fine-tune the pre-trained model and evaluate the prediction results utilizing the videos with more than S_p steps respectively on training and validation samples. We compared our method with previous ones on Table II using the $S_p = 3$ model. From the results, we can see that OCRL outperforms the previous methods on the prediction task with a different number of steps. In detail, OCRL(ii) has the best performance when total step S_i equals 4, 5, and 7; OCRL(i) works the best when $S_i = 6$. Compared to the model pre-trained by StepNCE in OCRL, InfoNCE [42]-based model only contains instance discrimination information, while UberNCE [46]-based model utilizes weakly supervised information from labels. However, neither of them considers the messages from temporal order. Moreover, the OCRL(i) model has comparable results with respect to OCRL(ii), which implies that the encoder learned by StepNCE can already generate reliable features for instructional video prediction without fine-tuning. Those facts have proven that step order provides a very powerful clue for understanding instructional tasks. An interesting fact is that prediction accuracy does not decrease monotonously when the total number of steps for the task is increased. The best prediction performance occurs when $S_i = 6$ on OCRL(i). The prediction difficulty of a certain task may not be related to its total step count, but to the specificity of its containing steps. For example, the task *make matcha tea* only contains

TABLE VI: **Average video prediction accuracy on COIN with different settings of loss function.** All the models are trained under the fine-tuned strategy. (ID: instance discrimination, TC: task consistency, OC: order consistency)

Method ($S_p = 3$)	**Accuracy(%)**
ID	32.66
ID + TC	47.97
ID + OC	23.91
ID + TC + OC (ours)	**60.00**

TABLE VII: **Pre-training accuracy on COIN with different choices of S_p.** S_i for each pre-training video clip x_i equals to S_p.

Choice of S_p	**Pre-training Accuracy(%)**
2 steps	34.34
3 steps	**48.51**
4 steps	47.63
5 steps	48.05

4 steps, but the whole process is nearly the same as task *make tea*. It is very hard to make out the difference before the final step *add milk* occurs. In contrast, the 7-step task *change bike tires* is very recognizable even only with the first step *unload the wheel* observed. This phenomenon implies that studying a task does not require abundant observations of steps (but proper ones) to acquire enough information for making predictions. Furthermore, we compare the OCRL approach to other methods under different choices of S_p. The results are shown in Table III. It can be seen that $S_p = 3$ OCRL(ii) model works better when $S_i = 5$, while $S_p = 4$ OCRL(ii) model and $S_p = 3$ OCRL(i) model are the most accurate ones when $S_i = 6$. This phenomenon implies that the prediction accuracy of different S_p-models is correlated to the total count of steps. The influence of choices on different S_p values will be further discussed in ablation studies.

In CrossTask, all the 2,588 primary-task videos are split into 2,296 training samples and 292 testing samples. We use the same pre-training and fine-tuning strategy as we use on COIN. We choose $S_p = 4$ model to evaluate on CrossTask instead of $S_p = 3$, the results are shown in Table IV. OCRL(ii) outperforms other methods under testing samples of all S_i values. The highest accuracy occurs when $S_i = 6$. Table V further demonstrates the prediction performance under different S_p values on CrossTask. We find out that $S_p = 3$ model works better when $S_i = 5$, while $S_p = 4$ model is more accurate when $S_i = 6$.

We can compare the results on COIN with the performances on CrossTask of our approach. It is worth noting that the label system in CrossTask is slightly different from the one in COIN. CrossTask does have fine labels for each video, which specify the task and its corresponding action steps. However, CrossTask does not have a unified action step label system, which means that the action contents are specified per video clip. The steps that happen in one video could be non-sequential due to the film editing reason, or the same task can have multiple ways and different permutations of steps to accomplish. In both cases, the order-consistency rule could be broken without a unified label system. Therefore, the performance gain of OCRL on CrossTask is not so much as it on COIN. Even so, our method can still obtain more accurate predictions compared to previous approaches, which further emphasizes the importance of information of sequential order.

D. Ablation Studies

We perform the ablation studies mainly on COIN. Several adjustments to our method have been conducted to measure the influence of different settings.

Defining positive and negative pairs. We conduct the experiments on the different definitions of positive/negative samples (Table VI). We build the basic structure of StepNCE loss using TC-rule and OC-rule based upon the concept of instance discrimination. The ID in Table VI refers to the instance discrimination NCE loss as shown in (1). ID+TC refers to TC-based NCE loss, with the positive and negative samples defined as follows:

$$\mathcal{P}_i = \{\phi(x_p) \mid y_p = y_i,\ \forall p \in [1, N]\} \tag{6}$$

$$\mathcal{N}_i = \{\phi(x_n) \mid y_n \neq y_i,\ \forall n \in [1, N]\} \tag{7}$$

where all the notations are consistent with (2) and (3). ID+OC refers to OC-based NCE loss, with the positive and negative samples defined as follows:

$$\mathcal{P}_i = \{\phi(x_p) \mid o_p = o_i,\ \forall p \in [1, N]\} \tag{8}$$

$$\mathcal{N}_i = \{\phi(x_n) \mid o_n \neq o_i,\ \forall n \in [1, N]\} \tag{9}$$

The expression of loss function for both ID+TC and ID+OC approaches is consistent with (4).

The results show that TC-rule and OC-rule are both essential to the task prediction objective. From the result of 'ID+TC', it can be learned that by adding task-consistency into instance discrimination loss, a 15.31% performance gain has been achieved. However, training the encoder only based on OC-rule would lead to bad results. From the result of 'ID+OC', the accuracy even decreased if only the order-consistency rule is added. This phenomenon provides further evidence for the ambiguity and complexity of step order. When the step order in a video is reversed, the semantic of its task is very likely to be changed as well. If no other constraints are added to the reversed video (*i.e.* the original task label is still used), it can lead to mistaken task information. The above view throws out the prediction-related information, thus it decreases the performance [53]. Meanwhile, this result can be considered as proof that step order clues are different from the frame order clues as used in [14]. To gain the best performance, we have to consider both consistencies into StepNCE. The result of our approach displays a significant performance gain from the instance discrimination loss (27.34%), which also proves the effectiveness of the two consistency rules.

Discussing on the choices of different S_p. The choice of S_p, number of steps, that our network preview to predict the tasks can affect final performance (Table III, V, VII). Although the input video length is identical, the number of peeked actions can reveal different quantities of information for the ongoing task, thus leading to unfairness for task predicting.

TABLE VIII: **Video prediction accuracy (%) under different training strategy.** The upper row refers to strategy (i), while the lower row refers to strategy(ii).

Methods	$S_p = 2$	$S_p = 3$
w/o fine-tuned	50.31	58.13
w/ fine-tuned	51.41	**60.00**

TABLE IX: **Task retrieval accuracy (%) .** We compare the retrieval accuracy using $k = 1, 5, 10$ under several different pre-training settings.

Method			**R@1**	**R@5**	**R@10**
S_p	TC	OC			
3	√	√	**43.06**	**64.80**	**74.31**
2	√	√	33.79	59.01	67.28
3	√	×	24.87	48.26	57.18
3	×	×	14.24	28.77	36.03

First, we compare the pre-training accuracy under different S_p to study the learning difficulty of different choices of S_p in Table VII. The accuracy here refers to the ratio of the pre-trained model to correctly differentiate positive and negative samples from the memory bank. The $S_p = 3, 4, 5$ models have similar pre-training accuracies, while $S_p = 2$ model has a very poor effect compared to other models. It proves that two given steps among a task provide insufficient information about step order. It is more common to see two steps rather than three can be reversed without changing the semantic of task. Thus, $S_p > 2$ models would be better choices for instructional video prediction. Further, in Table III and V, we find out that the task predicting performance also depends on the proper choices of S_p. The $S_p = 3$ model works better with shorter tasks ($S_i = 5$), while $S_p = 4$ model works better with longer tasks ($S_i = 6$). In intuition, larger S_p should lead to better prediction performances, since more step information is previewed by the model. However, the issue in instructional video task prediction is different from the one in action prediction. We do not need to observe as many steps as possible to make predictions. Furthermore, since our method used a fixed clip length of input videos, there is a balancing issue between step-wise information and within-step information. Larger S_p values will lead to a shorter step length. For example, the step length is $l_s = 10$ when $S_p = 3$, but reduces to $l_s = 8$ when $S_p = 4$. For shorter tasks, the step-wise information is not so complex. Under this circumstance, too many observed steps are not needed for making a prediction. Thus, we use smaller S_p to cover more information within each step. Whereas for longer tasks, a few observation steps may not provide enough information for correct predictions. In this case, a larger S_p value can achieve better performances. In addition, if a longer observation frame length is permitted, larger S_p values could be more beneficial than smaller S_p values.

Strategies for training the prediction network. According to the Implementation Details part, we adopt two different strategies for training the prediction network. We conduct experiments with different choices of training strategy, and calculate the average prediction accuracies in Table VIII. On both $S_p = 2$ and $S_p = 3$ models, the fine-tuned networks perform slightly better (1.1% and 1.87%) than the networks without fine-tuned. The conclusion is easy to be drawn since

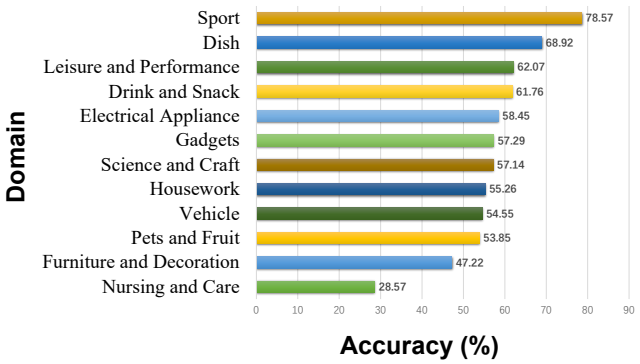

Fig. 7: **Average video prediction accuracy on COIN across different domains.**

TABLE X: **The prediction accuracy (%) on COIN with different temporal augmentations when $S_p = 3$.** The swapping operation swaps the order of the first two steps(segments) of a clip.

Methods	**Number of steps (size of test set)**			
($S_p = 3$)	4 (518)	5 (125)	6 (26)	7 (8)
w/o augmentations	38.6	47.6	57.7	**50.0**
steps-swapped	40.0	45.2	61.5	25.0
steps-reversed(ours)	**52.9**	**54.0**	**65.4**	**50.0**

strategy (ii) backpropagates gradients into the backbone, and updates the parameters of encoders. The key is that strategy (ii) only gains tiny improvements. When using strategy (i), the performances of the linear classifier only depend on the task-discriminating capacity of the feature encoder with partially observed steps. The comparable performance of strategy (i) implies that our pre-trained model can encode adequate task-related information into the video features. Besides, one drawback of the fine-tuning strategy is that it requires a larger-scale set of parameters to update, which makes it take a longer time to be trained.

Task retrieval for analyzing feature distinctiveness. To analyze the task-distinctiveness of our learned representations, we perform task retrieval on the encoders (Table IX and Fig. 6). Specifically, we use the nearest-neighbor (NN) retrieval approach to search for the most similar video clips. The videos in the testing set are used to query the k-nearest neighbors inside the training set. We evaluate the retrieval performance using the recall at k (R@k) metric, which refers to the top k ranking video clips. The retrieval performances are very useful metrics to evaluate the effectiveness of feature encoders. This retrieval study measures the task-discriminating capacity of our pre-trained encoders. In Table IX, the $S_p = 3$ model with both task-consistency and order-consistency rules obtains the best retrieval performance. Without order-consistency rule or both consistency rules, the accuracy drops steeply (-18.19% or -28.82%). From this result, we can

TABLE XI: **The prediction accuracy (%) on COIN using/not using background frames when $S_p = 3$.**

Methods	**Number of steps (size of test set)**			
($S_p = 3$)	4 (518)	5 (125)	6 (26)	7 (8)
w/ backgrounds	41.5	41.1	57.7	**50.0**
w/o backgrounds(ours)	**52.9**	**54.0**	**65.4**	**50.0**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2022.3149329, IEEE Transactions on Circuits and Systems for Video Technology

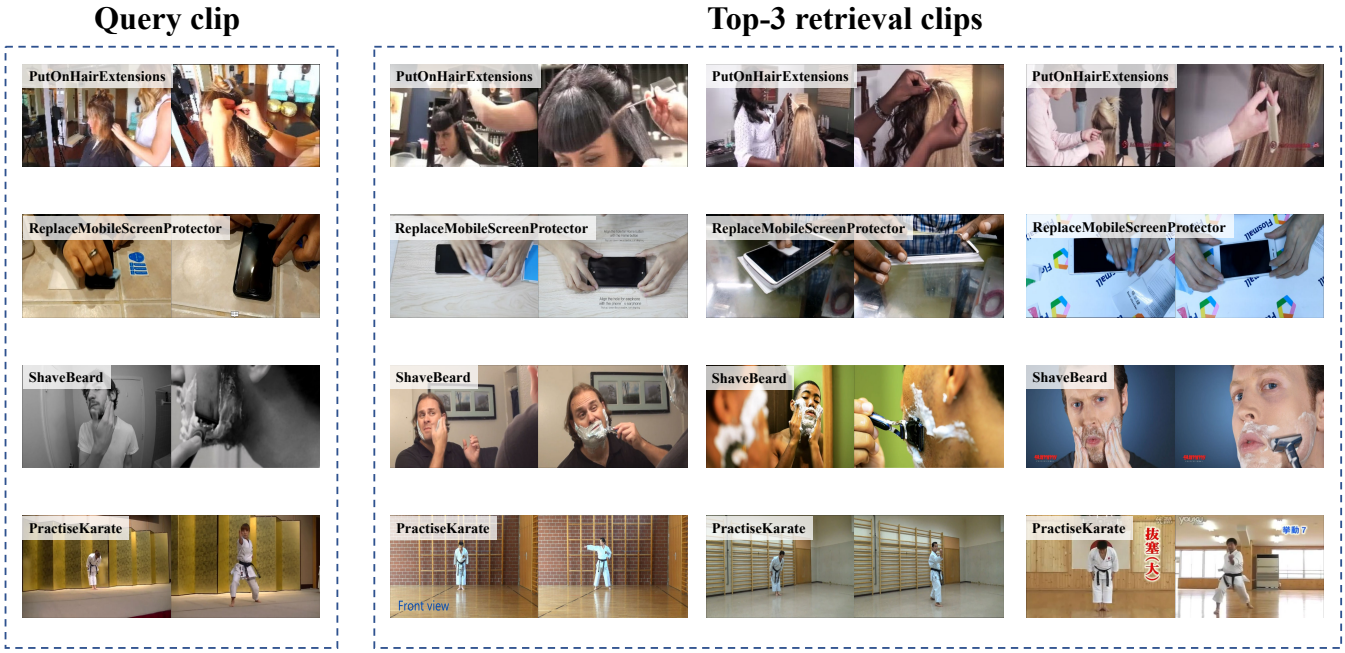

Fig. 6: **Visualization of task retrieval results with StepNCE.** In the illustration, not only task labels of the query clip are consistent with the retrieval results, but the consisted step labels are also equivalent.

see that StepNCE has the greatest ability to distinguish input videos from different tasks. Moreover, the $S_p = 2$ model with both consistencies also advances 8.92% on R@1 metric compared to the $S_p = 3$ model with only the task-consistency rule, which proves the usefulness of introducing the order-consistency rule. In Fig. 6, some visual examples using the $S_p = 3$ model with both consistency rules are displayed. Except for the capability of retrieving the video clips with the same task, the top-3 results also show that our pre-trained model retrieved the clip with the most similar component actions. For example, for the reference sample from task *practice karate*, all the performers in the retrieved clips play a punch after saluting. This phenomenon shows that our model retrieves the clips utilizing the information of component actions. It proves that our pre-trained model has the ability to learn the task information from component action steps, which is very useful when predicting the tasks from partially observed steps.

Predicting tasks across different domains. COIN dataset contains 12 domains of instructional videos. We conduct our prediction methods on videos from diverse domains (Fig. 7), to explore the sequentiality of tasks in different fields. We use the $S_p = 3$ model with both consistency rules to acquire the results. The prediction performances depend much on the sequentiality of domain actions. It is shown on the results that *Sport* and *Dish* domains gain the highest prediction accuracy with 78.57% and 68.92%. In a video clip of a sport-type task, the step actions are more closely related to each other both temporally and semantically. Moreover, sport-related and dish-related tasks have very distinct features in each step which implies the task information (e.g. scenes, or interactive objects). *Nurse and Care* and *Furniture and Decoratio* domains, however, score the lowest (28.57% and 47.22%) potentially because of the semantical ambiguity of actions and their sequential logic. To be specific, the above tasks can happen in various situations and different scenes, which increases the difficulty of task prediction. This study gives an insight into human cognition research on diverse practical domains.

Using different temporal augmentations. In our approach, we use the order-reversion operations to steps for the learning of OC-rule. There are other temporal operations, including step-swapping, which are not used in our approach. We conducted some experiments to test the different temporal augmentations in Table X. Here, the step-swapping operation is used by randomly swapping the first two steps out of the three observed steps. From the results, it can be seen that the steps-swapped model has little promotion from the base model when $S_i = 4, 6$ (1.4% and 3.8% respectively), but drops when $S_i = 5, 7$. The performance of the steps-swapped model is not so good as the steps-reversed model. The results show that step-swapping operation is not an applicable way to describe OC-rule compared to step-reversing operation. This phenomenon is intuitional, since swapping the first two steps may not affect the final task semantic of a certain observed instructional video clip of certain tasks. For example, when performing the task *making tea*, it is not essential whether the performer is putting the water in the teapot first or putting the tea first, but surely the hot tea cannot appear before the ingredients are added. In spite of this, the performances of the step-swapping model are better than the model without utilizing any information of temporal augmentation, which proves the efficiency of introducing the OC-based methods into the objective of task prediction.

The usage of background frames. Through our approach, we used trimmed videos for training and validation. However,

TABLE XII: **The prediction accuracy (%) on UTI #1 and UTI #2.** The experiments are conducted on the observation ratios of 0.1, 0.3 and 0.5.

Methods	UTI Set #1			UTI Set #2		
	0.1	0.3	0.5	0.1	0.3	0.5
IBoW [57]	15.0	30.0	65.0	18.3	35.0	45.0
DBoW [57]	15.0	45.0	70.0	18.3	41.7	51.7
MSSC [61]	18.3	60.0	70.0	21.7	50.0	71.7
Lan *et al.* [62]	35.0	68.3	83.1	33.3	56.7	78.3
MTSSVM [63]	38.3	66.7	78.3	31.7	60.0	73.3
AAC [64]	45.0	60.0	91.7	51.7	60.0	81.7
MMAPM [65]	46.7	70.0	78.3	36.7	61.7	75.0
PA-DRL [32]	49.3	76.7	91.7	41.7	63.3	83.3
RSPG+SVM [66]	51.7	78.3	91.7	46.7	68.3	85.0
AORAP Net [34]	45.1	67.3	**94.3**	48.5	70.0	**91.6**
OCRL (ours)	**63.3**	**81.6**	88.3	**63.3**	**75.0**	80.0

TABLE XIII: **The prediction accuracy (%) on BIT-Interaction.** The experiments are conducted on the observation ratios of 0.3, 0.5, 0.7 and 0.9.

Methods	BIT-Interaction			
	0.3	0.5	0.7	0.9
IBoW [57]	37.3	49.2	46.6	44.4
DBoW [57]	40.6	46.9	55.5	55.5
MSSC [61]	41.4	48.4	60.2	66.4
Lai *et al.* [69]	55.9	79.4	84.4	85.0
MTSSVM [63]	45.0	60.0	66.8	71.3
DeepSCN [68]	59.4	78.1	86.7	88.0
AAPNet [9]	64.8	80.5	88.3	91.4
Wu *et al.* [33]	58.6	81.3	89.1	86.7
RSPG+SVM [66]	71.3	87.0	88.6	90.2
AORAP Net [34]	**71.5**	**92.9**	**96.8**	**94.8**
OCRL (ours)	65.6	84.4	90.6	89.1

most of the instructional video contains abundant content with spoken narration and redundant visual content. For example, the performer in an instructional video usually describes the task details and the performing process at the very first beginning of the video, with no actions performed during this period. When the performer is making actions, some additional language interpretations may also be interluded between the steps. It is worth noting that our method is a pure visual-based approach for instructional task prediction. Thus, using the videos which are trimmed according to step information will be beneficial to verify our hypothesis. We apply the prediction experiments with regard to background frames in Table XI. The with-background method uses three randomly sampled segments from a complete video to form the input clips. We also conduct temporal-reversing operations on those clips to uniform with the OCRL approach. From Table XI, model without background frame as inputs works better than model with backgrounds when $S_i = 4, 5, 6$, and the two performances equals when $S_i = 7$. The results show that trimmed videos provide more action-related visual information, thus are more applicable to instructional video prediction.

E. Comparison with Other Prediction Methods

The performances of OCRL are also evaluated in some other action prediction datasets including UTI and BIT-Interaction. We compare OCRL with previous state-of-the-art methods including IBoW [67], DBoW [67], MSSC [61], MTSSVM [63], MMAPM [65], DeepSCN [68], AAC [64], PA-DRL [32], AAPNet [9], RSPG [66], AORAP Net [34], and approaches in Lai *et al.* [69], Lan *et al.* [62] and Wu *et al.* [33]. Most of the popular action recognition/prediction datasets are composed of videos with continuous single actions. For example, UCF101 [6] is a RGB-based dataset with 101 categories of action. However, the video contents in UCF101 mostly are pure repeated actions with no sequential information, which is inapplicable to our approach. Specifically, we choose UT-Interaction and BIT Interaction to evaluate our method for two reasons: (i) although each video clip in those datasets only contains a single violent or nonviolent action, the scenarios can usually be divided into three parts according to the pre-action scene and post-action scene, which are seen as three-component steps during pre-training stage; (ii) predicting crime scenes is meaningful with a wide range of applications. Furthermore, the prediction results can be used to evaluate the effectiveness of the StepNCE-based video representation learning model. Despite the dissimilarity of application scenarios for our method compared to conventional action prediction methods, OCRL shows comparable results on the conventional datasets.

UTI #1 and UTI #2. On the dataset of UTI #1 and UTI #2, we utilized the training settings in [57]. We use a 10-fold leave-one-out cross-validation during the evaluation process, by choosing 6 testing clips out of the 60 videos per validation. We pre-train the StepNCE model on the combination of two sets. Then, we train the prediction network separately on two sets, and do the evaluations with observation ratios of 0.3, 0.5, and 0.9 respectively. Because our method is specifically designed for instructional video prediction, which utilizes the step information to trim the video. To fit in our method, we evenly divide the videos into S_p clips under each observation ratio (during the pre-training stage we use fully-observed information). We evaluate the prediction performances on the testing set with observation ratios of 0.1, 0.3, and 0.5 respectively. From Table XII, we find out that OCRL performs better in very-early video prediction from the results with observation ratios of 0.1 and 0.3. This result implies that the StepNCE-based feature encoder efficaciously captures the relationship between pre-action scenes and action labels. When the observation ratio reaches 0.5, the performance of OCRL can also obtain comparable results compared to 'AORAP Net', 'RSPG+SVM' and 'PA-DRL' approaches. The performance is not better since we did not design specific structures for action prediction, while previous methods were mostly aimed at solving it.

BIT-Interaction. On the BIT-Interaction dataset, we follow the training settings in [58]. Among the 400 video clips, 68% of them (272 clips) are partitioned as the training set, while the other 32% (128 clips) are utilized as the testing set. We use the same data sampling strategy as we do in the UT-Interaction dataset. We fine-tune our pre-trained model on the training set. Then the prediction network is trained using the training set, and evaluated on the testing set with observation ratios of 0.3, 0.5, 0.7, and 0.9 respectively (Table XIII). The evaluation results show that OCRL gains a competitive performance with the previous state-of-the-art methods. 'AORAP Net' has stronger performance than our approach, since it

TABLE XIV: **The activity recognition accuracy (%) on ActivityNet v1.2.**

Methods	ActivityNet v1.2	
	Top1-Acc (%)	Top5-Acc (%)
S3D (scratch) [55]	26.9	58.5
InfoNCE-based(*ss*) [42]	29.8	61.9
UberNCE-based(*ws*) [46]	**46.0**	**73.6**
OCRL(*ws*) (ours)	43.5	72.3

plants an additional Observation Ratio Regression Module (we only utilize a single linear classifier) while using a pre-trained TSN [70] as the feature encoder (we use S3D as the backbone). When the observation ratio is low (0.3 and 0.5), our approach performs not so well as the 'RSPG+SVM' approach which utilizes the skeleton data from raw RGB videos. Skeleton-based approaches are more suitable to the videos that contain more complete and more consistent human poses while the backgrounds are irrelevant, which matches the case of BIT and UTI datasets. Even so, our approach obtains a higher prediction performance at 0.7 observation ratio than 'RSPG+SVM'. The accuracy drops at a very high observation ratio (0.9) partly due to the finite observed duration in each video. Performances can be improved with a longer input frame length.

F. Comparative Study on General Video Recognition

The previous experimental studies have proved that our proposed OCRL approach has the ability to capture the task semantics from multiple sequential action steps of instructional videos. We also conducted experiments on the activity recognition dataset to verify if the order constraints also exist in general video datasets. ActivityNet v1.2 is a large-scale activity recognition dataset, of which each video only contains a single human activity. We trim the videos according to the provided activity positional annotations, and divide each trimmed video into three equal parts. We randomly sample the three clips from the above divided parts as the three-component steps for the pre-training stage. The results are shown in Table XIV. From the results, we find that OCRL does not outperform the UberNCE-based approach, which indicates that order constraints are not beneficial, or may even be harmful to the understanding of single-action semantics in general video datasets. For single human action, permuting the order of several action segments does not affect the action semantics. For example, within the human action of *dribbling*, whatever how the order of action segments is altered does not affect human's judgment on what the player is acting in the video clip. In other words, order constraints are uncorrelated to object semantics for single action recognition in general video datasets. However, order constraints are proved to be crucial to instructional video task semantics in the previous studies.

V. Conclusions

In this paper, we have focused on the challenging task of instructional video prediction. We proposed an Order-Constrained Representation Learning (OCRL) approach to extract visual features based on the semantic complicacy among action steps and between steps and tasks. We analyzed the semantic significance of step order in instructional videos, which has a very compact relationship with task information. We proposed two assumptions, *task consistency (TC)* and *order consistency (OC)*, based on the consistency analysis of instructional video. Under the two consistency hypotheses, we established a new contrastive loss, StepNCE, in which the positive samples are defined only when two consistency rules are both achieved. The semantic sequentiality of different action steps with regard to a certain task is exploited by StepNCE-based encoders. The pre-trained encoder is then fine-tuned to make predictions on the instructional video task by partly observed video clips. OCRL method gains a higher-level understanding of video semantics. Finally, We prove the strong capability of our approach for instructional video task prediction on COIN, CrossTask, UTI, BIT, and ActivityNet v1.2 datasets.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62125603, Grant U1813218, and in part by a grant from the Beijing Academy of Artificial Intelligence (BAAI).

References

[1] J.-B. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien, "Unsupervised learning from narrated instruction videos," in *CVPR*, 2016, pp. 4575–4583.

[2] D.-A. Huang, J. J. Lim, L. Fei-Fei, and J. Carlos Niebles, "Unsupervised visual-linguistic reference resolution in instructional videos," in *CVPR*, 2017, pp. 2183–2192.

[3] A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman, "End-to-end learning of visual representations from uncurated instructional videos," in *CVPR*, 2020, pp. 9879–9889.

[4] D. Zhukov, J.-B. Alayrac, R. G. Cinbis, D. Fouhey, I. Laptev, and J. Sivic, "Cross-task weakly supervised learning from instructional videos," in *CVPR*, 2019, pp. 3537–3545.

[5] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *CVPR*, 2017, pp. 6299–6308.

[6] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[7] A. Rasouli, "Deep learning for vision-based prediction: A survey," *arXiv preprint arXiv:2007.00095*, 2020.

[8] Y. Tang, D. Ding, Y. Rao, Y. Zheng, D. Zhang, L. Zhao, J. Lu, and J. Zhou, "Coin: A large-scale dataset for comprehensive instructional video analysis," in *CVPR*, 2019, pp. 1207–1216.

[9] Y. Kong, Z. Tao, and Y. Fu, "Adversarial action prediction networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 539–553, 2018.

[10] X. Wang, J.-F. Hu, J.-H. Lai, J. Zhang, and W.-S. Zheng, "Progressive teacher-student learning for early action prediction," in *CVPR*, 2019, pp. 3556–3565.

[11] J. Huang, N. Li, T. Li, S. Liu, and G. Li, "Spatial–temporal context-aware online action detection and prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 8, pp. 2650–2662, 2019.

[12] J. Weng, X. Jiang, W.-L. Zheng, and J. Yuan, "Early action recognition with category exclusion using policy-based reinforcement learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 12, pp. 4626–4638, 2020.

[13] B. Fernando, H. Bilen, E. Gavves, and S. Gould, "Self-supervised video representation learning with odd-one-out networks," in *CVPR*, 2017, pp. 3636–3645.

[14] D. Wei, J. J. Lim, A. Zisserman, and W. T. Freeman, "Learning and using the arrow of time," in *CVPR*, 2018, pp. 8052–8060.

[15] D. Xu, J. Xiao, Z. Zhao, J. Shao, D. Xie, and Y. Zhuang, "Self-supervised spatiotemporal learning via video clip order prediction," in *CVPR*, 2019, pp. 10 334–10 343.

[16] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, "Temporal relational reasoning in videos," in *ECCV*, 2018, pp. 803–818.

[17] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips," in *ICCV*, 2019, pp. 2630–2640.

[18] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, "Videobert: A joint model for video and language representation learning," in *ICCV*, 2019, pp. 7464–7473.

[19] A. Kukleva, H. Kuehne, F. Sener, and J. Gall, "Unsupervised learning of action classes with continuous temporal embedding," in *CVPR*, 2019, pp. 12 066–12 074.

[20] M. Xu, J.-M. Pérez-Rúa, V. Escorcia, B. Martinez, X. Zhu, B. Ghanem, and T. Xiang, "Boundary-sensitive pre-training for temporal localization in videos," *arXiv preprint arXiv:2011.10830*, 2020.

[21] D. Zhukov, J.-B. Alayrac, I. Laptev, and J. Sivic, "Learning actionness via long-range temporal order verification," in *ECCV*, 2020, pp. 470–487.

[22] J. Wang, W. Wang, and W. Gao, "Predicting diverse future frames with local transformation-guided masking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 12, pp. 3531–3543, 2018.

[23] H. Gao, H. Xu, Q.-Z. Cai, R. Wang, F. Yu, and T. Darrell, "Disentangling propagation and generation for video prediction," in *ICCV*, 2019, pp. 9006–9015.

[24] Y.-H. Kwon and M.-G. Park, "Predicting future frames using retrospective cycle gan," in *CVPR*, 2019, pp. 1811–1820.

[25] X. Lin, Q. Zou, X. Xu, Y. Huang, and Y. Tian, "Motion-aware feature enhancement network for video prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 2, pp. 688–700, 2020.

[26] S. Li, J. Fang, H. Xu, and J. Xue, "Video frame prediction by deep multi-branch mask network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 4, pp. 1283–1295, 2020.

[27] H. Gunes, M. A. Nicolaou, and M. Pantic, "Continuous analysis of affect from voice and face," in *Computer Analysis of Human Behavior*. Springer, 2011, pp. 255–291.

[28] M. Valstar, J. Gratch, B. Schuller, F. Ringeval, D. Lalanne, M. Torres Torres, S. Scherer, G. Stratou, R. Cowie, and M. Pantic, "Avec 2016: Depression, mood, and emotion recognition workshop and challenge," in *ACMW*, 2016, pp. 3–10.

[29] Z. Du, S. Wu, D. Huang, W. Li, and Y. Wang, "Spatio-temporal encoder-decoder fully convolutional network for video-based dimensional emotion recognition," *IEEE Trans. Affect. Comput.*, 2019.

[30] P. Felsen, P. Agrawal, and J. Malik, "What will happen next? forecasting player moves in sports videos," in *ICCV*, 2017, pp. 3342–3351.

[31] M. S. Aliakbarian, F. S. Saleh, M. Salzmann, B. Fernando, L. Petersson, and L. Andersson, "Encouraging lstms to anticipate actions very early," in *ICCV*, 2017, pp. 280–289.

[32] L. Chen, J. Lu, Z. Song, and J. Zhou, "Part-activated deep reinforcement learning for action prediction," in *ECCV*, 2018, pp. 421–436.

[33] X. Wu, R. Wang, J. Hou, H. Lin, and J. Luo, "Spatial–temporal relation reasoning for action prediction in videos," *Int. J. Comput. Vis.*, vol. 129, no. 5, pp. 1–22, 2021.

[34] C. Liu, Y. Gao, Z. Li, C. Du, F. Liu, and X. Shi, "Action prediction network with auxiliary observation ratio regression," in *ICME*, 2021, pp. 1–6.

[35] Z. Luo, B. Peng, D. A. Huang, A. Alahi, and F. F. Li, "Unsupervised learning of long-term motion dynamics for videos," in *CVPR*, 2017, pp. 7101–7110.

[36] A. Diba, V. Sharma, L. V. Gool, and R. Stiefelhagen, "Dynamonet: Dynamic action and motion network," in *ICCV*, 2019, pp. 6191–6200.

[37] Y. Yao, C. Liu, D. Luo, Y. Zhou, and Q. Ye, "Video playback rate perception for self-supervisedspatio-temporal representation learning," in *CVPR*, 2020, pp. 6547–6556.

[38] D. Kim, D. Cho, and I. S. Kweon, "Self-supervised video representation learning with space-time cubic puzzles," *AAAI*, vol. 33, pp. 8545–8552, 2019.

[39] J. Wang, J. Jiao, L. Bao, S. He, Y. Liu, and W. Liu, "Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics," in *CVPR*, 2019, pp. 4006–4015.

[40] I. Misra, C. L. Zitnick, and M. Hebert, "Shuffle and learn: Unsupervised learning using temporal order verification," *ECCV*, pp. 527–544, 2016.

[41] H. Y. Lee, J. B. Huang, M. Singh, and M. H. Yang, "Unsupervised representation learning by sorting sequences," in *ICCV*, 2017, pp. 667–676.

[42] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748,*, 2018.

[43] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020, pp. 1597–1607.

[44] T. Han, W. Xie, and A. Zisserman, "Video representation learning by dense predictive coding," in *ICCVW*, 2019, pp. 1483–1492.

[45] ——, "Memory-augmented dense predictive coding for video representation learning," in *ECCV*, 2020, pp. 312–329.

[46] ——, "Self-supervised co-training for video representation learning," in *NeurIPS*, 2020, pp. 5679–5690.

[47] J. Wang, J. Jiao, and Y. H. Liu, "Self-supervised video representation learning by pace prediction," in *ECCV*, 2020, pp. 504–521.

[48] T. Yao, Y. Zhang, Z. Qiu, Y. Pan, and T. Mei, "Seco: Exploring sequence supervision for unsupervised representation learning," in *AAAI*, vol. 35, no. 12, 2021, pp. 10 656–10 664.

[49] R. Qian, T. Meng, B. Gong, M.-H. Yang, H. Wang, S. Belongie, and Y. Cui, "Spatiotemporal contrastive video representation learning," in *CVPR*, 2021, pp. 6964–6974.

[50] B. Korbar, D. Tran, and L. Torresani, "Cooperative learning of audio and video models from self-supervised synchronization," in *NeurIPS*, 2018, pp. 7763–7774.

[51] A. Piergiovanni, A. Angelova, and M. S. Ryoo, "Evolving losses for unsupervised video representation learning," in *CVPR*, 2020, pp. 133–142.

[52] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *AISTATS*, 2010, pp. 297–304.

[53] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What makes for good views for contrastive learning?" in *NeurIPS*, 2020, pp. 6827–6839.

[54] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020, pp. 9729–9738.

[55] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *ECCV*, 2018, pp. 305–321.

[56] D. Zhukov, J. B. Alayrac, R. G. Cinbis, D. Fouhey, I. Laptev, and J. Sivic, "Cross-task weakly supervised learning from instructional videos," in *CVPR*, 2019, pp. 3537–3545.

[57] M. S. Ryoo and J. Aggarwal, "Ut-interaction dataset, icpr contest on semantic description of human activities (sdha)," in *ICCVW*, 2010, pp. 270–285.

[58] Y. Kong, Y. Jia, and Y. Fu, "Learning human interaction by interactive phrases," in *ECCV*, 2012, pp. 300–313.

[59] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *CVPR*, 2015, pp. 961–970.

[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015, pp. 1–15.

[61] Y. Cao, D. Barrett, A. Barbu, S. Narayanaswamy, H. Yu, A. Michaux, Y. Lin, S. Dickinson, J. Mark Siskind, and S. Wang, "Recognize human activities from partially observed videos," in *CVPR*, 2013, pp. 2658–2665.

[62] T. Lan, T.-C. Chen, and S. Savarese, "A hierarchical representation for future action prediction," in *ECCV*, 2014, pp. 689–704.

[63] Y. Kong, D. Kit, and Y. Fu, "A discriminative model with multiple temporal scales for action prediction," in *ECCV*, 2014, pp. 596–611.

[64] Z. Xu, L. Qing, and J. Miao, "Activity auto-completion: Predicting human activities from partial videos," in *ICCV*, 2015, pp. 3191–3199.

[65] Y. Kong and Y. Fu, "Max-margin action prediction machine," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1844–1858, 2015.

[66] L. Chen, J. Lu, Z. Song, and J. Zhou, "Recurrent semantic preserving generation for action prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 231–245, 2020.

[67] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in *ICCV*, 2011, pp. 1036–1043.

[68] Y. Kong, Z. Tao, and Y. Fu, "Deep sequential context networks for action prediction," in *CVPR*, 2017, pp. 1473–1481.

[69] S. Lai, W.-S. Zheng, J.-F. Hu, and J. Zhang, "Global-local temporal saliency action prediction," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2272–2285, 2017.

[70] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks for action recognition in videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2740–2755, 2018.

Muheng Li received the B.S. degree from the Department of Engineering Physics, Tsinghua University, Beijing, China, in 2020, where he is currently pursuing the M.S. degree with the Department of Automation. His current research interests concentrate on computer vison, especially human behaviour understanding, and multimodal video understanding.

Lei Chen received the B.S. degree in the Qiushi Honers college, and the Ph.D degree with the school of electrical and information engineering, Tianjin University, China, in 2013 and 2020. His current research interest lies in human behaviour understanding for computer vision. He has 5 papers are published in top journals and conferences including CVPR, ECCV, PR and TCSVT. He serves as a regular reviewer member for a number of journals and conferences, e.g., TIP, TCSVT, CVPR, ICCV, AAAI and so on.

Jiwen Lu (M'11-SM'15) received the B.Eng. degree in mechanical engineering and the M.Eng. degree in electrical engineering from the Xi'an University of Technology, Xi'an, China, in 2003 and 2006, respectively, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2012. He is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing, China. His current research interests include computer vision and pattern recognition. He has authored/co-authored over 300 scientific papers in these areas, where 100+ of them are IEEE Transactions papers and 100+ of them are CVPR/ICCV/ECCV papers. He serves the Co-Editor-of-Chief of the Pattern Recognition Letters, an Associate Editor of the IEEE Transactions on Image Processing, the IEEE Transactions on Circuits and Systems for Video Technology, the IEEE Transactions on Biometrics, Behavior, and Identity Science, and Pattern Recognition. He is a member of the Image, Video and Multidimensional Signal Processing Technical Committee, the Multimedia Signal Processing Technical Committee and the Information Forensics and Security Technical Committee of the IEEE Signal Processing Society, and a member of the Multimedia Systems and Applications Technical Committee and the Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems Society, respectively. He was a recipient of the National Outstanding Youth Foundation of China Award. He is a Fellow of the IAPR and a senior member of the IEEE.

Jianjiang Feng received the B.Eng. and Ph.D. degrees from the School of Telecommunication Engineering, Beijing University of Posts and Telecommunications, China, in 2000 and 2007, respectively. From 2008 to 2009, he was a Post-Doctoral Researcher with the PRIP Laboratory, Michigan State University. He is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing. His research interests include fingerprint recognition and computer vision.

Jie Zhou (M'01-SM'04) received the BS and MS degrees both from the Department of Mathematics, Nankai University, Tianjin, China, in 1990 and 1992, respectively, and the PhD degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology (HUST), Wuhan, China, in 1995. From then to 1997, he served as a postdoctoral fellow in the Department of Automation, Tsinghua University, Beijing, China. Since 2003, he has been a full professor in the Department of Automation, Tsinghua University. His research interests include computer vision, pattern recognition, and image processing. In recent years, he has authored more than 300 papers in peer-reviewed journals and conferences. Among them, more than 100 papers have been published in top journals and conferences such as the IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Image Processing, and CVPR. He is an associate editor for the IEEE Transactions on Pattern Analysis and Machine Intelligence and two other journals. He received the National Outstanding Youth Foundation of China Award. He is a Fellow of the IAPR and a senior member of the IEEE.