International Journal of Computer Vision (2023) 131:3272–3288
https://doi.org/10.1007/s11263-023-01850-6

Dynamic Context Removal: A General Training Strategy for Robust Models on Video Action Predictive Tasks

Xinyu Xu[1] · **Yong-Lu Li**[1] · **Cewu Lu**[1]

Received: 10 October 2022 / Accepted: 12 July 2023 / Published online: 13 August 2023

Abstract
Predicting future actions is an essential feature of intelligent systems and embodied AI. However, compared to the traditional recognition tasks, the uncertainty of the future and the reasoning ability requirement make prediction tasks very challenging and far beyond solved. In this field, previous methods usually care more about the model architecture design but little attention has been put on how to train models with a proper learning policy. To this end, in this work, we propose a simple but effective training strategy, Dynamic Context Removal (DCR), which dynamically schedules the visibility of context in different training stages. It follows the human-like curriculum learning process, i.e., gradually removing the event context to increase the prediction difficulty till satisfying the final prediction target. Besides, we explore how to train *robust* models that give consistent predictions at different levels of observable context. Our learning scheme is *plug-and-play* and easy to integrate widely-used reasoning models including Transformer and LSTM, with advantages in both effectiveness and efficiency. We study two action prediction problems, i.e., Video Action Anticipation and Early Action Recognition. In extensive experiments, our method achieves state-of-the-art results on several widely-used benchmarks.

Keywords Dynamic Context Removal · Video Action Anticipation · Early Action Recognition · Robustness

1 Introduction

A comprehensive understanding of action sequences, e.g., `open the can before pouring water out`, is a basic ability of humans. We usually know how to take multiple action steps to achieve a final target and are easy to reason out the next action based on the past context. It puts new requirements on embodied AI as advanced intelligence should possess the ability to understand the action order and predict the next one. Thus, action prediction matters. It also serves as a support for many applications like autonomous driving (Alvarez et al., 2020; Rasouli et al., 2019) and human-robot interaction (Koppula & Saxena , 2015; Ryoo et al., 2015), where the predictions on pedestrians and users are essential.

With the rapid evolution of deep learning techniques, the comprehensive understanding and analysis of human action videos attract attention in edging research. In the traditional recognition field, modern video models (Carreira & Zisserman , 2017; Fan et al., 2021; Feichtenhofer et al., 2019; Lin et al., 2019; Simonyan and Zisserman , 2014; Tran et al., 2018, 2019; Wang et al., 2016, 2018) leverage spatiotemporal modeling to learn both spatial patterns and temporal logics and achieve significant progresses in many video *recognition* problems (Damen et al., 2021; Goyal et al., 2017; Kay et al., 2017). Besides, there is also a growing interest in action *prediction* problems (Damen et al., 2018, 2021; Kuehne et al., 2014; Li et al., 2018; Stein & McKenna , 2013). Similarly, they both expect systems to discriminate the existing actions in videos. Differently, the observed video segment given for systems shifts in action prediction problems, while action recognition systems have all the contents of actions from videos.

However, due to the temporal misalignment between visual observation and target action semantics, action prediction problems are much more challenging than action

Communicated by Yoichi Sato.

✉ Yong-Lu Li
yonglu_li@sjtu.edu.cn

✉ Cewu Lu
lucewu@sjtu.edu.cn

Xinyu Xu
xuxinyu2000@sjtu.edu.cn

[1] Shanghai Jiao Tong University, Shanghai 200240, China

recognition. It can hardly be treated as a simple classification like video recognition for some reasons. First, the spatial configuration which deep neural networks learn in prediction problems is biased towards the supervision of future action labels, leading to the inaccurate representation of the current visual observation. Second, the observation has a gap with the full action event, which challenges the high-level reasoning ability of models especially in the long-term dense action prediction setting (Ke et al., 2019; Sener et al., 2020).

Past works on action prediction usually propose elegant model architecture designs (Dessalene et al., 2021; Furnari & Farinella , 2020; Gao et al., 2017; Girdhar & Grauman , 2021; Gu et al., 2021; Sener et al., 2020; Wu et al., 2021), including LSTM variants and Transformer variants to learn temporal action logic from past contents and reason the future action. Though such methods achieve improvements, they still face performance bottlenecks on challenging benchmarks (Damen et al., 2018, 2021; Kuehne et al., 2014; Li et al., 2018; Stein & McKenna , 2013). In this work, apart from the design, we try to explore an effective and universal strategy for training predictive models, which is generally overlooked before. Existing works may directly map visual observation into the action label space (Girdhar & Grauman , 2021; Wu et al., 2021) or apply simple multitasking (using different anticipation times) progress in training (Furnari & Farinella , 2020). Instead, we propose Dynamic Context Removal (DCR). Predicting future action based on different anticipation times or observation rates can be seen as multitasking. DCR follows the curriculum learning (Bengio et al., 2009) insight to train easy tasks first and then hard ones. It employs sufficient auxiliary context at first then removes redundant context for better adaptation to the difficult evaluation settings. Figure 1 gives an intuitive example. DCR achieves a finer controlled learning procedure to tell what to learn in different training phases, greatly differing from naive multitasking and making more sense.

In this work, we study two problems: Video Action Anticipation and Early Action Recognition, which are most practical in action prediction. Video Action Anticipation is to predict the action label in the future but the observable contents have a gap (anticipation time τ_a) before the action occurs. τ_a is usually a fixed parameter on standard benchmarks (Damen et al., 2018, 2021). For this single-task, DCR interpolates the intermediate training difficulties to offer a *smooth* curriculum learning path. Early Action Recognition is to predict the action label when the action goes on but does not end, within the limited partial observation of the segment. It usually has multiple tasks in testing, i.e., evaluating model performance at different observation rates. We propose an advanced version of DCR on multiple tasks, which provides a more detailed consideration on balancing different tasks and keeping consistent predictions. We make a more detailed

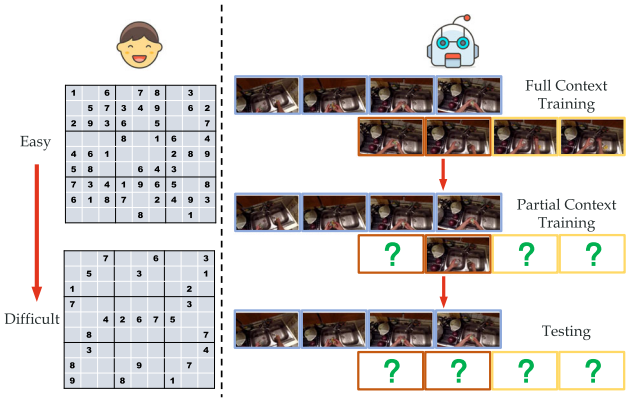


Fig. 1 Revisiting learning curriculums in the classical Sudoku game, a kid starts with an easy Sudoku game of more observation (hints) and then gets taught a harder level of less observable numbers. This reveals the curriculum learning process of how humans learn to reason in the physical world. In this work, we are inspired by learning Sudoku and build action anticipation models with similar curriculum designs. We leverage extra auxiliary frames in training but dynamically schedule their visibility to gradually strengthen the reasoning ability of models

analysis of model performances when removing observable context and new metrics are set up.

Our training strategy DCR is *plug-and-play* and can boost multiple temporal predictive architectures. Considering the high potential of Transformer (Vaswani et al., 2017) in recent years, we take it by default. It starts with the order-aware pre-training phase, specific to Transformer, where we leverage the permutation-invariant property of attention mechanism and apply frame order as a self-supervised signal to do the pre-training. Next, we conduct the action prediction training. Our systems leverage mask modeling to reconstruct masked frame representations and dynamically schedule the visibility of auxiliary context. For single-task Video Action Anticipation and multi-task Early Action Recognition, our implementations are slightly different to facilitate two tasks respectively. This learning paradigm conforms to how humans learn, i.e., the curriculum learning insight. We also validate DCR on LSTM (Hochreiter & Schmidhuber , 1997) to prove its generalizable effect.

We conduct experiments and analyses on four widely-used Video Action Anticipation benchmarks: EPIC-KITCHENS-100 (Damen et al., 2021), EPIC-KITCHENS-55 (Damen et al., 2018), EGTEA GAZE+ (Li et al., 2018), 50-Salads (Stein & McKenna , 2013) and two Early Action Recognition benchmarks: EPIC-KITCHENS-55 (Damen et al., 2018), EGTEA GAZE+ (Li et al., 2018). Our training strategy turns out to be effective and achieves state-of-the-art results on all benchmarks, even using small-size parameter-efficient models. Moreover, we believe the proposed dynamic and adaptive learning paradigm can pave the way for more complex and challenging temporal prediction problems.

Overall, our contribution includes: (1) We propose a simple but effective learning strategy, DCR, which advances the effectiveness and efficiency of practical temporal modeling architectures including Transformer and LSTM on action prediction tasks. (2) We propose the order-aware pre-training for Transformer to carry out unsupervised pre-training using frame sequential order as self-supervision. (3) We introduce the perspective of consistent prediction at different observation rates in Early Action Recognition, where new metrics and training techniques are proposed. (4) We achieve the state-of-the-art on several widely-used Video Action Anticipation and Early Action Recognition benchmarks.

Note that, a prior version of DCR is published in CVPR 2022 (Xu et al., 2022), which only contains content about Video Action Anticipation. In this new version, we adapt DCR to Early Action Recognition and study prediction robustness in multiple tasks. To this end, we make new contributions to propose a novel method, evaluation metrics and achieve SOTA performances.

2 Related Work

Video Action Anticipation is to predict the future action label by observing a video clip with time τ_a before it occurs. It is required both in third-person (Kuehne et al., 2014; Stein & McKenna , 2013) and egocentric (Damen et al., 2018, 2021; Ke et al., 2019; Li et al., 2018; Liu et al., 2020) scenarios. It supports a wide range of applications including intelligent robots (Koppula & Saxena , 2015; Ryoo et al., 2015) and wearable devices. It used to have different formulations such as dense action anticipation, but we consider predicting the next action in this work. Previous methods proposed various neural architectures including LSTM variants (Farha et al., 2018; Furnari & Farinella , 2020; Furnari et al., 2018; Gao et al., 2017; Jain et al., 2016; Wu et al., 2021) and attention variants (Girdhar & Grauman , 2021; Gu et al., 2021; Sener et al., 2020). In the early work, Vondrick et al. (2016) propose an unsupervised representation learning paradigm to connect the feature of the present and future for action anticipation problems. Li et al. (2018) jointly model action anticipation with human gaze in egocentric videos. Later, Furnari and Farinella (2020) propose a classic RULSTM architecture with modularity attention which achieves strong results. Sener et al. (2020) attempt to anticipate action with different aggregations on the past. Some other works utilized extra knowledge like next active object (Furnari et al., 2017) and hand motion (Dessalene et al., 2021) to anticipation action. AVT (Girdhar & Grauman , 2021) leverages a causal Transformer to model action anticipation in the *seq2seq* manner.

Early Action Recognition is to recognize actions from an incomplete observation of the whole action segment as early as possible (Ryoo , 2011). It is similar to the action anticipation problem as they both expect models to reason out the unobserved state of action frames. Thus, some classic networks (Dessalene et al., 2021; Furnari & Farinella , 2020; Sener et al., 2020) on action anticipation can easily migrate to Early Action Recognition well. But differently, Early Action Recognition usually requires model performance under different levels of observation ratios, which is an explicit form of multitasking compared to the τ_a-fixed action anticipation problem. Hu et al. (2019) introduce an early prediction framework based on soft regression paradigm. Wang et al. (2019) propose a teacher-student distillation framework as a progressive learning paradigm in this field. Pang et al. (2019) apply a unified encoder-decoder framework to jointly model the bi-directional video dynamics. IGFormer (Pang et al., 2022) proposes Interaction Graph Transformer for skeleton-based early recognition. A recent work ERA (Foo et al., 2022) replaces 3D Conv by expert retrieval and assembly for prediction. In this work, we study the previously ignored robustness property of models at different observation ratios, *i.e.,* making consistent predictions among high and low observable context ratios. We validate the effects of DCR techniques under new metrics.

Video Sequential Order Modeling has been exploited in many works. Srivastava et al. (2015) propose unsupervised learning techniques to learn generalized representation in video sequences. Zhou and Berg (2015) explore two simple tasks of pairwise ordering and future prediction in egocentric videos. Kong et al. (2017) model sequential context relation to advance the recognition performance on part video observations. Misral et al. (2016) propose a new perspective of video sequence as to verify whether the order is correct in learning. In our work, we leverage the permutation invariant property of self-attention mechanism and utilize sequential order as an extra signal to perform self-supervised learning.

Vision Transformer gains much popularity recently, with a trend to exceed the classic convolution architecture in many visual tasks. Transformer (Vaswani et al., 2017) family originally raises in the language community, then permeates into the vision domain (Dosovitskiy et al., 2021) including video related tasks (Arnab et al., 2021; Fan et al., 2021; Wang et al., 2018). It can be inserted as attention blocks (Wang et al., 2018; Wu et al., 2019) into traditional video models as well as construct pure attention-based video recognition architecture (Arnab et al., 2021; Fan et al., 2021). In the field of Video Action Anticipation, Transformer architecture can be directly used in temporal reasoning via causal attention (Girdhar & Grauman , 2021).

Curriculum Learning is proposed by Bengio et al. (2009). It is motivated by the learning procedure of humans, from easy to hard. It can be implemented via the schedule of category loss weights (Kumar et al., 2010), data sampling (Li et al., 2017), or other difficulty measurement (Zhang et al., 2020).

This simple principle works well in many fields including language understanding (Bengio et al., 2009), transfer learning (Weinshall et al., 2018) and more (Kumar et al., 2010; Li et al., 2017; Zhang et al., 2020)). For the language reasoning task, previous work (Cirik et al., 2016) also validates its effectiveness when doing baby-step short-term reasoning first. In our work, the richness of auxiliary context determines the easiness of tasks. We schedule the context removal to obey the *easy-to-hard* principle of curriculum learning.

3 Approach

We introduce our method in this section. First, the detailed formulations of two video action prediction problems are introduced in Sect. 3.1. Then, an overview of our system is described in Sect. 3.2, which can integrate any temporal predictive architectures. For the Transformer, we propose an additional order-aware pre-training (Sect. 3.3) to learn temporal dynamics in the *full context* mode, which is not used for LSTM. Next, we train prediction models for Video Action Anticipation and Early Action Recognition in the *partial context* mode. For the single-task Video Action Anticipation in Sect. 3.4, we apply a random context interpolation strategy. The auxiliary context in the anticipation time is randomly selected to help the prediction, but probability decreases in training with a decreasing task easiness. We set two objectives including prediction loss and reconstruction loss. For the multi-task Early Action Recognition in Sect. 3.5, we propose a finer-grained task selection method on improving the prediction ability within the limited observation. An additional consistency constraint is used to alleviate the performance drop when the observable context is removed.

3.1 Problem Formulation

We briefly introduce the settings of two video action prediction problems, i.e., Video Action Anticipation and Early Action Recognition, which are illustrated in Fig. 2. In our data processing scheme, we adopt different colors for different temporal segments to indicate their roles. Blue ones are the consistent video observation. Red ones are the main field with dynamic visibility. Yellow ones are used for prediction purposes. We use mask models to reconstruct yellow frame representations based on blue and part of red frames.

Video Action Anticipation is to predict the future action label before it occurs. There is a time gap between the observation and the action segment. It is called anticipation time, expressed as τ_a. The anticipation time is usually fixed as τ_a = 1 s on standard benchmarks and leaderboards (Damen et al., 2018, 2021; Girdhar & Grauman , 2021; Li et al., 2018; Sener et al., 2020). Thus, we only care about 1-second anticipation performance. We call this as single-task in evaluation.

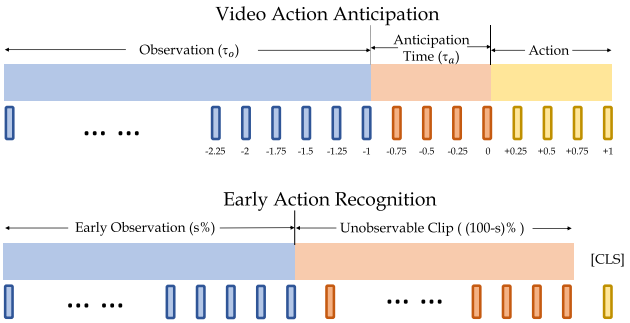


Fig. 2 The general setting of the two video action prediction problems. Blue, red and yellow frames indicate different fields in data processing (Color figure online)

Another important parameter is τ_o, which denotes the length of the observable clip. Usually, τ_o is not restricted and any possible choices are permitted. We sample frames at 4 fps following (Furnari & Farinella , 2020). There are $4\tau_o/4/4$ frames assigned with blue/red/yellow (Fig. 2) respectively.

Early Action Recognition requires an early prediction of the action label from partial observation. The first s (observation rate) action clip is observable for action reasoning models, while the remaining is not. We sample 40 frames from each action clip, where the first $s\%$ are the early observation in blue and the last $(100 - s)\%$ in red are not visible in test time. Following previous work (Furnari & Farinella , 2020), we care about model performances with different levels of observation rates, i.e., $s\% = 12.5\%, 25\%, 37.5\%, 50\%, 62.5\%, 75\%, 87.5\%, 100\%$. Since the multitasking nature of Early Action Recognition, it differs from the τ_a-fixed anticipation problem. It has multiple tasks in evaluation. In our work, the robustness of models at the decreasing of observation rates is explored. Notably, to achieve the prediction purpose, an additional learnable `[CLS]` token in yellow (Fig. 2) is used for classification.

3.2 Overview

We present an overview of the system we build in Fig. 3. Assume we sample K frames for our model, then we start from K pre-extracted frame representations as $(x_1, x_2, \ldots, x_K)$, in the reverse chronological order. Each frame x_i is assigned with a binary mask $\beta_i \in \{0, 1\}$, determining its visibility. The mask is dynamically scheduled in different phases of training (Sects. 3.4 and 3.5), but we strictly mask out auxiliary frames in the test time to meet the testing setting. Our system aims to reconstruct masked frame representations based on the remaining context, especially the yellow frames for label prediction purposes.

Specifically, we project frame feature $x_i (1 \leq i \leq K)$ into a latent space, where a reasoning model $\mathcal{R}$ performs to reason out the masked frames based on visible information. Then, a

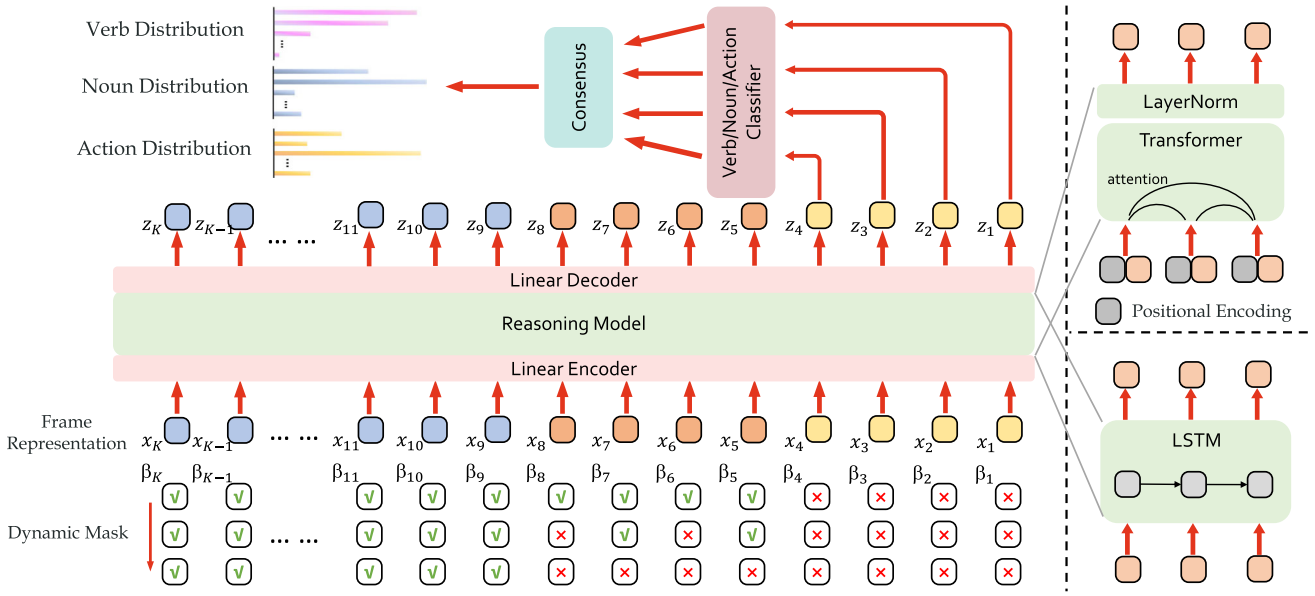


Fig. 3 An overview of our system, which aims to use observable contents to reconstruct masked frame representations. The temporal reasoning model in the system can have choices including Transformer and LSTM. Frames in blue/red/yellow are used for constant observation, dynamic visibility, and action classification. The core motivation of DCR is to schedule the frame visibility in a curriculum learning manner, with more auxiliary frames at first but dynamically removed as the training goes on. To achieve this, we apply dynamic masks at different training times, i.e., more observations when training begins but fewer observations later (Color figure online)

linear decoder maps frames back to the original dimension. For our goal of the reconstructions from the reasoning model, we denote them as $(z_1, z_2, \ldots, z_K)$ respectively. It is formulated as $(z_1, z_2, \ldots, z_K) = \mathcal{R}(x_1, \beta_1, x_2, \beta_2, \ldots, x_K, \beta_K)$. The reconstructions of yellow frames in Fig. 2 are used for predicting the label. In Video Action Anticipation, the last 4 frames $z_i (1 \leq i \leq 4)$ are in the action occurrence and they will be sent to the classifier to give predictions. But in Early Action Recognition, we employ an extra learnable `[CLS]` token to make the role. For EPIC-KITCHENS series (Damen et al., 2018, 2021) which also require marginalized verb/noun class prediction on their test server, we use verb/noun/action three classifiers on the top but only apply single action classifier for other datasets. In the test time, predictions on these yellow frames are averaged to make a consensus (Wang et al., 2016) as the final result.

Noticeably, our training strategy is flexible and can be used for widely-used reasoning models, including Transformer (Vaswani et al., 2017), LSTM (Hochreiter & Schmidhuber , 1997) etc. In this paper, considering the exceeding potential of Transformer in recent works, we use it for most experiments by default to compare against prior arts but also show our superiority in LSTM based results. A small difference between Transformer and LSTM is about tackling masked frames. Masks are more practical for Transformer based applications (Devlin et al., 2019), so we directly assign zero values for the input. But for the recurrent LSTM structure, it is more sensitive about the latest observation and zero values lead to the smooth prediction, thus we copy the masked frames using the latest visible (not masked) one, similar to the unrolling mechanism (Furnari & Farinella , 2020).

3.3 Order-Aware Pre-training

As explained above, we use Transformer as the default architecture. To fully exploit its potential, we propose a novel order-aware pre-training to learn the temporal logic in the *full context mode*. An ideal video reasoning model should be able to automatically understand the sequential order of action steps as well as visual frames. Thus, we leverage a sorting task in the pre-training phase to model the video dynamics, which greatly boosts predictive tasks in the next stage.

In this stage, we notice that Transformer is a permutation invariant architecture without explicit positional encoding (Vaswani et al., 2017). Thus, we use temporal positions as signals to supervise the training and expect models to automatically recognize the order of input sequence, which implies the understanding of temporal logic among context.

We propose our self-supervised pre-training technique called order-aware pre-training. All frames except the `[CLS]` token illustrated in Fig. 2 are used in training. In another word, the mask $\beta_i (1 \leq i \leq K)$ consistently equals 1. Without explicit integration of positional encoding, they are directly sent into the Transformer after the linear encoder. Then we compute the cosine similarity between i-th Trans-

former output tokens and the j-th positional encoding as $\widetilde{s}_{i,j}$, which is followed by Softmax to probability space as $\widetilde{p}_{i,j}$.

A pre-defined similarity label is required to supervise the training. The most naive choice is to use a diagonal matrix as similarity to treat it as a separate classification problem. However, time series is continuous, and it would be much better to assign soft labels. To this end, we follow (Hayat et al., 2019) to define similarity with Gaussian affinity. The similarity $s_{i,j}$ of positional encoding at time i and the frame feature at time j is measured as

$$s_{i,j} = \exp\left(-\frac{(i-j)^2}{\sigma^2}\right), \tag{1}$$

where σ is the bandwidth of Gaussian and we set $\sigma = 5$ in our experiments. Then the similarity is used to supervise the pre-training. We minimize the cross entropy loss L in Eq. 2 with the Gaussian soft labels:

$$L = \sum_{i=1}^{K}\sum_{j=1}^{K} -s_{i,j}\log(\widetilde{p}_{i,j}). \tag{2}$$

The order-aware pre-training not only learns the video dynamics in the context but also provides a refinement of the positional encoding. It is an aggregated context agnostic representation of the whole dataset and is more suitable for the masked frames in the following training phase. Generally, the technique is motivated by the permutation-invariant property of self-attention and utilizes sequence permutation as self-supervised signals. We believe it can advance a wider range of Transformer based sequence modeling tasks, just like the success of masked language model (Devlin et al., 2019).

3.4 A Training Route for Single-Task Prediction

For the single-task Video Action Anticipation problem, we use interpolation context to connect anticipation on $\tau_a = 1$ s and $\tau_a = 0$ s and gradually increase training task difficulty by context removal. Apparently, smaller τ_a means an easier task. Therefore, we set an easy curriculum ($\tau_a = 0$ s) at the beginning of the training, where all auxiliary frames in the anticipation time are observable. As the training goes on, auxiliary frames are randomly removed, controlled by a decreasing easiness factor. It interpolates a novel training route to adapt models to the more difficult $\tau_a = 1$ s anticipation task.

3.4.1 Random Context Masking

Since the goal of our system is to reconstruct masked future frames based on the visible context, we use *partial context* in training and dynamically schedule the context visibility. In

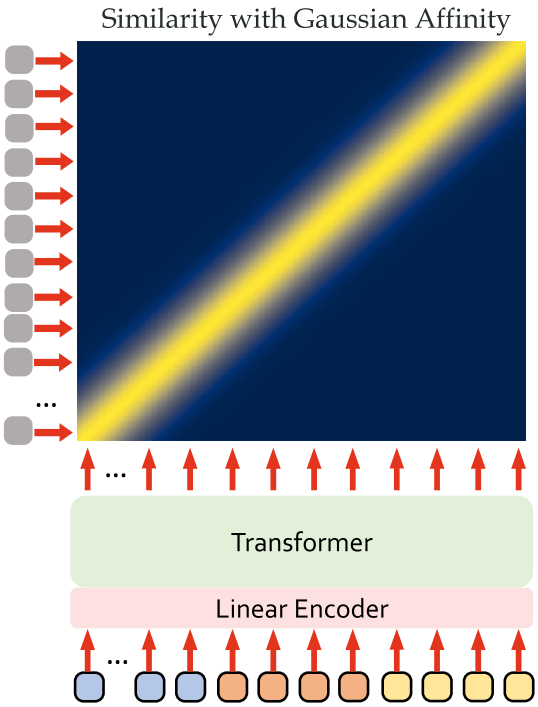


Fig. 4 The order-aware pre-training for the permutation-invariant attention. We remove the positional encoding on the input side but force the model to automatically understand the order of video sequences. It is trained by connecting the frame with its corresponding position to meet the pre-defined similarity

training epoch e, we formulate an easiness factor $T_e \in [0, 1]$, determining the easiness of curriculum in training. As it starts with an easy curriculum, we have $T_1 = T_2 = 1$ initialliy. Then, as the training goes on, we decrease the easiness T_e of the curriculum for our system and present a more difficult prediction task gradually.

T_e indicates the probability of employing additional auxiliary frames to assist prediction. In this phase, we consistently mask out yellow frames during action occurrence, as $\beta_i = 0 (i \leq 4)$. Notably, the numerical setting is attributed to the frame sampling scheme in Sect. 3.1, i.e., $4\tau_o/4/4$ frames are in blue/red/yellow (defined in Sect. 3.1 and Fig. 2) respectively for Video Action Anticipation. Yellow frames are not directly sent for model input but serve as the supervision of the feature reconstructions. We find the direct utilization of action frames harms classifier performance and reasoning ability in our experiments. On the contrary, past observation, i.e., blue frames in Fig. 2, are always visible at any time, as $\beta_i = 1 (i \geq 9)$. The red frames in the median field are the main ones for designing different curriculums, i.e., dynamic visibility in the training. They assist past observation to reconstruct the anticipated future frames but are dynamically removed and determined by the decreasing T_e. For $5 \leq i \leq 8$, we uniformly sample variable ρ_i in [0,1], as $\rho_i \in U(0, 1)$. The frame x_i is visible only when ρ_i is smaller than the easiness factor T_e. It also means these

frames have a probability T_e to be visible. It is formulated as $\beta_i = \mathbf{1}[T_e > \rho_i](5 \leq i \leq 8)$, where $\mathbf{1}[*]$ indicates the truth of statement and returns binary value. Generally, we obtain β series in Eq. 3:

$$\beta_i = \begin{cases} 1 & i \geq 9 \quad \text{(blue frames)}, \\ \mathbf{1}[T_e > \rho_i] & 5 \leq i \leq 8 \quad \text{(red frames)}, \\ 0 & i \leq 4 \quad \text{(yellow frames)}. \end{cases} \tag{3}$$

Empirically, we design a fine-grained local curriculum scheduling method in this work. Though a global schedule of T_e like linear or exponential may also work well in some scenarios (Sect. 4.6), we find it is sensitive to hyper-parameters tuning and not very convenient. To this end, we empirically apply a more specific easiness schedule. In each iteration, mask $\{\beta_1, \beta_2, \ldots, \beta_K\}$ is generated for a video clip. Assume $k(k \geq 5)$ is smallest for $\beta_k = 1$ then $x_1, \ldots, x_{k-1}$ are what we need to anticipate. We use the error of the 1-second future to measure the quality of the reconstruction:

$$\begin{aligned} & Q = \|x_{k-4} - z_{k-4}\|_2, \\ s.t. \quad & k = argmin[\beta_k = 1]. \end{aligned} \tag{4}$$

A memory bank is used to store the reconstruction quality for each case. It serves as a criterion to define the *easiness* in the next epoch. We have $T_1 = T_2 = 1$ at the start, but simply schedule easiness T_e using the decline of Q in Eq. 5, with extra boundaries $\gamma_{min} = 0.95, \gamma_{max} = 1$ on the decreasing factor. $\texttt{Trunc}(\cdot)$ is the truncation function. In this case, the rapid decline of Q represents a well-learned state of models, thus we decrease easiness faster. The boundaries are used to stabilize easiness scheduling and guarantee the diversity of curriculums in different training stages:

$$\frac{T_e}{T_{e-1}} = \texttt{Trunc}\left(\frac{Q_{e-1}}{Q_{e-2}}, \gamma_{min}, \gamma_{max}\right). \tag{5}$$

3.4.2 Learning Objective

We use two objectives to supervise the training process. One is about the predictive result of the next action class L_{cls}, while the other is the reconstruction loss of masked frames L_{rec}.

Prediction loss is used to supervise the prediction of yellow frames in the action segment. We adopt the cross entropy loss. We apply random label smoothing (Szegedy et al., 2016) for the loss. Though Camporese et al. (2021) study different label smoothing designs in anticipation, we find simple random smoothing already works well in the predictive tasks. This is mainly attributed to the advantages of random label smoothing on maintaining the uncertainty of the future and suppressing overfitting. As for z_i, action classifier gives prediction $p_i^1, p_i^2, \ldots, p_i^C$, where C is the number of categories. Then, the action prediction loss L_{cls}^A can be formulated in Eq. 6, where y is the ground truth label, w_y is the class loss weight from class distribution and ϵ is the factor of label smoothing:

$$L_{cls}^A = \sum_{i=1}^{4} -(1-\epsilon) w_y \log(p_i^y) - \sum_{c=1}^{C} \frac{\epsilon}{C} \log(p_i^c). \tag{6}$$

For datasets that require marginalized verb/noun predictions additionally, we compute verb/noun prediction loss similarly as L_{cls}^V, L_{cls}^N. The prediction loss is made as $L_{cls} = L_{cls}^V + L_{cls}^N + L_{cls}^A$. For datasets only with an action classifier on the top, we have $L_{cls} = L_{cls}^A$.

Reconstruction loss is to teach our model to reason out the masked frames based on the remaining context, just like the role of masked language prediction (Devlin et al., 2019). We expect the output representation of our reasoning model close to the original frame. Thus we simply use mean square error following (Girdhar & Grauman , 2021) in Eq. 7 as feature-level supervision:

$$L_{rec} = \sum_{i=1}^{K} (1 - \beta_i) * \|z_i - x_i\|_2. \tag{7}$$

Considering different scales and roles of two losses, we apply a weighted summation to obtain the total loss L_{total}:

$$L_{total} = \lambda_{cls} L_{cls} + \lambda_{rec} L_{rec}. \tag{8}$$

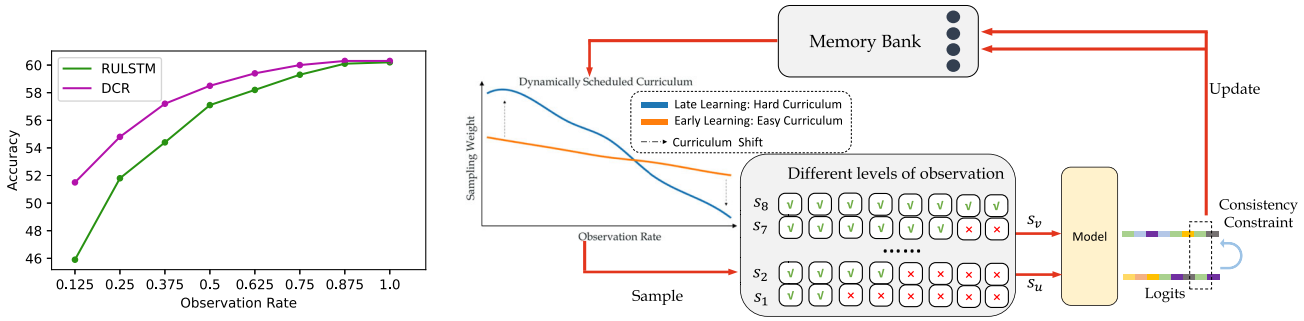
where λ_* are different weights for different loss items.

3.5 Towards Robust Multi-task Prediction

In the above Video Action Anticipation problem, we show interpolating extra context and gradually removing them in training helps the prediction. But we may wonder how our models preserve the predictive ability when frames are gradually removed. We explore this problem in Early Action Recognition, with our intention to keep trained models more robust on multiple tasks, i.e., giving robust predictions at different levels of observation rates. Figure 5 (a) gives an example. Our method and the baseline have similar performances given the full observation, but our method suffers less from the removed context and keeps robust within the limited observation.

3.5.1 Sampling Multiple Tasks

We set Early Action Recognition problem at different observation rates, i.e., from 12.5% to 100% in steps of 12.5%, as $s_1\%, s_2\%, \ldots, s_8\%$ respectively. Following the curriculum



(a) (b)

Fig. 5 **a** Comparison between the model we trained with the baseline on EGTEA GAZE+. DCR is more robust against the missing context as two methods have similar performance given full observation but our model suffers less at low observation rates. **b** The pipeline of our training procedure in Early Action Recognition problem. We acquire a distribution prior from the memory bank and sample two tasks of different difficulties. They are trained with the consistent prediction constraint to increase model robustness on different tasks. Predictions are used to update the sampling prior to make training curriculums shift from easy to hard gradually

learning insight, it should start with the easier task of more observation but adapt to the harder task gradually.

Our training pipeline is illustrated in Fig. 5 (b). For each action clip, we store a weight vector $w_k (k = 1, 2, \ldots, 8)$ in the memory bank, where w_k is the corresponding weight for sampling training task $s_k\%$. In each iteration, we use normalized $w_k / \sum_l w_l$ as the probabilistic distribution and sample two training tasks $s_u\%$ and $s_v\%$. Let $1 \leq u < v \leq 8$, which means $s_v\%$ has more visible context as the easy task and s_u is harder. For observation rates $s_u\%$, $s_v\%$, we generate their corresponding frame masks β_u, β_v. They are separately sent to our reasoning system to give the predicted results in two branches.

Initially, we have $w_k = 1 (k = 1, 2, \ldots, 8)$. This implies all tasks are uniformly sampled in the early stage of training, regardless of the observation rate. However, the weight vector for sampling is dynamically scheduled in different stages of training. It is mainly controlled by the prediction error. In epoch e, given two sampled tasks $s_u\%$, $s_v\%$ and their corresponding β_u, β_v, our reasoning system outputs the predicted probabilities p_u^y, p_j^v for the ground truth action class y while their errors are $1 - p_u^y$, $1 - p_v^y$ respectively. Notably, we use the 1-second future reconstruction as a measurement of reasoning ability in Sect. 3.4.1. But we cannot find a unified feature-level measurement in Early Action Recognition since the reconstruction may not be applied in some extreme cases like $s_8 = 100\%$. To this end, We use the predictive error in Eq. 9 to represent the reasoning ability of our model on a specific task:

$$\begin{aligned} Err_u &= 1 - p_u^y, \\ Err_v &= 1 - p_v^y. \end{aligned} \tag{9}$$

For each task s_k and its weight w_k used in sampling, we hold its updating strategy in the momentum manner in Eq. 10. If s_k is sampled in epoch $e - 1$, then the next w_k is weighted by its old value and the predictive error Err_k, with respect to η, an updating parameter fixed as $\eta = 0.8$ in our implementation. w_k equals 1 initially, but will converge to the predicting ability of models on task s_k in the end. Thus, harder tasks at less observation rates have larger predictive errors are assigned a larger probability in the sampling procedure. The system gradually leans to the harder tasks in training, which follows our motivation.

$$(w_k)_e = \begin{cases} \eta(w_k)_{e-1} + (1-\eta)(Err_k)_{e-1} & k \in \{u, v\}, \\ (w_k)_{e-1} & \text{otherwise.} \end{cases} \tag{10}$$

3.5.2 Consistent Prediction Constraint

We add an extra learning objective L_{con} on the consistent predictions between two tasks. Let $1 \leq u < v \leq 8$. We intend a model to give robust predictions though more frames are masked for s_u, close predicted results to the easier task s_v. Thus, we propose the consistent prediction constraint in the knowledge distillation manner. It uses the prediction from s_v to supervise the learning of s_u. We formulate the action consistency loss in Eq. 11:

$$L_{con}^A = \sum_{c=1}^{C} -p_v^c \log(p_u^c). \tag{11}$$

Notably, the gradient from p_v^c is stopped and only the harder task is trained. Similar to L_{cls}, we make $L_{con} = L_{con}^V + L_{con}^N + L_{con}^A$ for EPIC-KITCHENS series while $L_{con} = L_{con}^A$ for others.

We add the supplement L_{con} constraint and then the total loss is formulated in Eq. 12, where λ_* weights different loss items:

$$L_{total} = \lambda_{cls} L_{cls} + \lambda_{rec} L_{rec} + \lambda_{con} L_{con}. \quad (12)$$

4 Experiments

4.1 Datasets and Metrics

EPIC-KITCHENS-100 (EK100) (Damen et al., 2021) is currently the largest dataset to support action predictive tasks. It has 700 long videos of 100 h about egocentric cooking activities. Each action class in EK100 consists of a verb and a noun. Totally, there are 97 verbs and 300 nouns, leading to 4053 action compositions. There are 89,977 action segments whose labels are aggregated from unique narrations. The dataset splits into train/validation/test sets with a ratio of 75:10:15. The train and validation sets are publicly released but the test set is only able to be queried on the online server. The main metric for evaluation is recall@5, a class-aware metric to avoid the long-tail bias of action distribution. Besides, the authors also provide a tail action subset and an unseen participants subset to highlight the generalization performance of models.

EPIC-KITCHENS-55 (EK55) (Damen et al., 2018) is an earlier version of EK100. As a subset, it contains 432 videos in 55 h. There are 39,596 action segments, each assigned with a verb and noun class. It includes 125 verbs and 352 nouns in total. We follow the split of (Damen et al., 2018; Furnari & Farinella , 2020). The metric for evaluation is Top-1/5 accuracy.

EGTEA GAZE+ (EG+) (Li et al., 2018) is another egocentric dataset for the joint modeling of action and gaze. We only use its action learning part. It contains 19 verbs, 51 nouns, and 106 action compositions. There are 10,325 segments in 86 videos annotated with action labels. We report Top-5 accuracy and class-mean recall@5 over 3 standard official splits provided by the authors.

50-Salads (50S) (Stein & McKenna , 2013) is a widely used third-person video dataset about salad preparation. It's a relatively smaller dataset than the previous ones as it only has nearly 0.9K action segments. And differently, its action class can't be marginalized into verb and noun. We follow (Farha et al., 2018; Girdhar & Grauman , 2021; Sener et al., 2020) to use the 17-class coarse version of action annotation labels. We report Top-1 accuracy over 5 standard official splits provided by the authors.

In Video Action Anticipation experiments, we follow previous works (Damen et al., 2018, 2021; Furnari & Farinella , 2020; Girdhar & Grauman , 2021) to set τ_a = 1 s in all datasets. This setting is also shared for all baselines in a fair comparison. In Early Action Recognition, we evaluate models at different observation rates, i.e., from 12.5% to 100% in steps of 12.5%. Each observation rate s_i corresponds to a Top-1 accuracy score $a_i (i = 1, 2, \ldots, 8)$. We use three more metrics to give a unified evaluation, i.e., Avg, Std, APD. The first two are the mathematical calculation of the average value and the standard deviation of sequence $\{a_i \|_{i=1}^{8}\}$. APD is short for average performance drop, formulated as $\sum_{i=1}^{8} \left(\max_{1 \le j \le 8} a_j - a_i \right)/8$. Avg evaluates the general prediction performance while the others evaluate the robustness of model predictions at different observation levels, the lower the better.

4.2 Baselines

We compare DCR to several competitive approaches including FN (De Geest and Tuytelaars , 2018), vanilla LSTM (Hochreiter & Schmidhuber , 1997), RL (Ma et al., 2016), EL (Jain et al., 2016), DMR (Vondrick et al., 2016), ATSN (Damen et al., 2018), MCE (Furnari et al., 2018), FHOI (Liu et al., 2020), RULSTM (Furnari & Farinella , 2020), AutoBanks (Sener et al., 2020), ImagineRNN (Wu et al., 2021), Ego-OMG (Dessalene et al., 2021), TransAction (Gu et al., 2021), AVT (Girdhar & Grauman , 2021). Please refer to supplementary information for more details about baselines.

4.3 Implementation Details

Backbone We adopt different types of features (RGB appearance, Optical Flow and Object distribution) from different backbones, including ViT (Dosovitskiy et al., 2021), TSN (Wang et al., 2016), TSM (Lin et al., 2019) and irCSN-152 (Tran et al., 2019). More details can be found in supplementary information.

Observation In Video Action Anticipation, we set observation time τ_o = 10 s for EPIC-KITCHENS (Damen et al., 2021, 2018) series and 50 S (Stein & McKenna , 2013), but τ_o = 5 s for EG+ (Li et al., 2018). Longer observation requirement is mainly because of the larger data scale for EPIC-KITCHENS and longer action duration on average for 50 S.

Head Network Due to the high potential of Transformer in recent works, We use a 6-layer, 16-head Transformer encoder model (Vaswani et al., 2017) on 1024 dimensions optimized by AdamW (Loshchilov & Hutter , 2019) as the default reasoning architecture. In addition, our training strategy also boosts LSTM. We also conduct experiments using 1-layer, 1024-dimensional LSTM (Hochreiter & Schmidhuber , 1997) optimized by SGD. They are both trained from scratch.. For more hyper-parameter settings like learning rate schedule, batch size, and loss weight, please refer to supplementary information.

4.4 Results of Video Action Anticipation

First, we report single model performances as well as their trainable parameters on EPIC-KITCHENS series in Tables 1

Table 1 Single branch anticipation results on EK100 validation set

Method	Backbone	Verb	Noun	Action	# Params
RGB					
RULSTM	TSN	*27.5*	*29.0*	*13.3*	*19.7M*
AVT	TSN	*27.2*	*30.7*	*13.6*	*303.9M*
AVT	irCSN-152	25.5	28.1	12.8	409.6M
AVT	ViT*	28.7	32.3	14.9	383.8M
DCR (LSTM)	TSN	*27.9*	*28.0*	*14.5*	*14.1M*
DCR (LSTM)	TSM	28.4	28.5	15.2	20.2M
DCR	TSN	*31.0*	*31.1*	*14.6*	*78.2M*
DCR	TSM	**32.6**	**32.7**	**16.1**	84.3M
Flow					
RULSTM	TSN	19.1	16.7	7.2	19.7M
AVT	TSN	20.9	16.9	6.6	303.9M
DCR (LSTM)	TSN	21.6	15.3	7.8	14.1M
DCR	TSN	**25.9**	**17.6**	**8.4**	78.2M
Obj					
RULSTM	FRCNN	17.9	23.3	7.8	14.5M
AVT	FRCNN	18.0	**24.3**	8.7	298.8M
DCR (LSTM)	FRCNN	16.1	19.6	7.5	10.1M
DCR	FRCNN	**22.2**	24.2	**9.7**	74.2M

The backbone marked with * denotes end-to-end training
Numbers in bold are to highlight best performances under their settings

and 2 for a fair comparison. The baseline parameters are recorded from their public checkpoints. Our models have approximate parameters (numerical counts in Tables 1 and 2) except different dimensions of input spaces.

On EK100 validation set in Table 1, using the most widely-used RGB-TSN backbone (in italic), our LSTM version DCR is slightly lighter than classic RULSTM while the Transformer version is nearly a quarter of AVT because of the half network width. However, our models consistently perform better, especially for the default Transformer model, which has 3.8%/1.0% performance gains over AVT on verb/action respectively. Additionally, a more effective TSM backbone directly helps DCR to outperform the end-to-end trained AVT by a 1.2% margin at a lower expense. We scatter the performance and parameter size of RGB-input models in Fig. 6. Apparently, our models are in the upper left corner, indicating advantages in both effectiveness and efficiency. Besides, on flow and obj modality, our DCR also outperforms previous works. Especially for the flow, we have 5.0% and 1.2% performance gains on verb and action respectively.

Next, for results on EK55 validation set in Table 2, our DCR with RGB-TSN backbone also exceeds all baselines (italic) in a fair comparison. To our surprise, previous method applies a 12-layer deep Transformer on irCSN-152 backbone to achieve the best single model performance, but our light model easily outperforms it with a 2.3% gain on Top-5 score (in underline). The stronger TSM backbone further improves

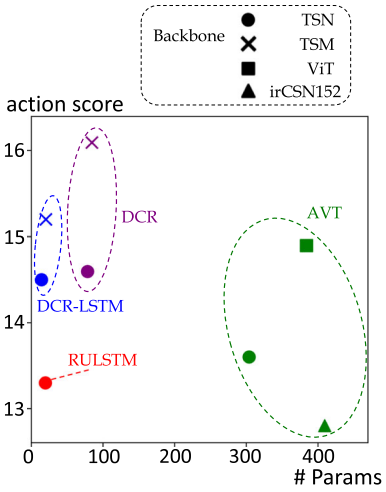


Fig. 6 Score versus Size

Table 2 Single branch anticipation results on EK55 validation set

Method	Backbone	Top-1	Top-5	# Params
RGB				
RULSTM	TSN	*13.1*	*30.8*	*18.5M*
ActionBanks	TSN	*12.7*	*28.6*	*112.9M*
AVT	TSN	*13.1*	*28.1*	*302.6M*
AVT	ViT*	12.5	30.1	382.8M
AVT	irCSN-152	14.4	31.7	603.2M
DCR	TSN	*13.6*	*30.8*	*78.2M*
DCR	irCSN-152	15.1	**34.0**	82.0M
DCR	TSM	**16.1**	33.1	82.0M
Flow				
RULSTM	TSN	8.7	21.4	18.5M
ActionBanks	TSN	8.4	19.8	112.9M
DCR	TSN	**8.9**	**22.7**	78.2M
Obj				
RULSTM	FRCNN	10.0	29.8	13.2M
ActionBanks	FRCNN	10.2	29.1	52.5M
DCR	FRCNN	**11.5**	**30.5**	74.2M

The backbone marked with * denotes end-to-end training
Numbers in bold are to highlight best performances under their settings

Top-1 action score by 1.7% over AVT. Besides, our method also achieves competitive results on flow and obj modalities, with 1.3% and 0.7% gains over prior arts respectively.

Since our training strategy focuses on the head reasoning network, apples-to-apples comparisons fairly verify the contribution of DCR in training effective anticipation models at a lower expense. It clearly paves the way for further research.

We also make model ensembles to give more accurate predictions on these benchmarks and compare to state-of-the-art methods. Despite previous work may use modality attention (Furnari & Farinella , 2020) or apply an extra Transformer to aggregate multi-modal tokens (Gu et al.,

Table 3 Anticipation result ensemble on EK100

Method	Validation		Test			
	Overall	Unseen	Tail	Overall	Unseen	Tail
RULTSM	14.0	14.1	11.1	11.2	9.7	7.9
ActionBanks	14.7	14.5	11.8	12.6	10.5	8.9
TransAction	16.6	13.8	15.5	13.4	10.1	11.9
AVT	15.9	11.9	14.1	16.7	12.9	13.8
DCR	**18.3**	**14.7**	**15.8**	**17.3**	**14.1**	**14.3**

Numbers in bold are to highlight best performances under their settings

2021), we simply use late-fusion results and DCR still shows superiority. Details about late fusion are in supplementary information.

We report results on EPIC-KITCHENS series in Tables 3 and 4. On the validation sets, we follow AVT to use *rgb+obj* fusion and it outperforms baselines, i.e., 1.7% performance gain on the whole EK100 and 3.6% Top-5 action accuracy gain on EK55. The competitions on the online testing leaderboard are more challenging. We make an ensemble using models trained with *train+val* data. Our method outperforms previous works on most branches. In Table 3, we achieve 0.7% improvement on EK100 overall and 1.2% improvement on its unseen subset. In Table 4, on Top-5 accuracy, we achieve 2.0% improvement on EK55 S1 and 0.5% on EK55 S2. We do not perform the prior art only on EK55 Top-1 accuracy, this is mainly because the competitive baseline Ego-OMG (Dessalene et al., 2021) adds delicate annotations of hand segmentation and active objects to learn intermediate knowledge representation, which helps anticipation in unseen environments.

In Table 5, we report results on EGTEA GAZE+, another popular egocentric anticipation benchmark. We use TSN feature on RGB and Optical Flow modalities following RULSTM to conduct this experiment. The final result is the late fusion of two branches. Surprisingly, DCR has 1.5% and 2.5% performance gains over all baselines on Top-5 accuracy and Recall@5 respectively, establishing the new *state-of-the-art*.

Table 5 Anticipation results on EG+

Method	Top-5	c.m. Recall@5
DMR	55.7	38.1
ATSN	40.5	31.6
MCE	56.3	43.8
TCN	58.5	47.1
ED	60.2	54.6
RL	62.7	52.2
EL	63.8	55.1
RULSTM	66.4	58.6
DCR	**67.9**	**61.1**

Numbers in bold are to highlight best performances under their settings

In Table 6, we report results on 50-Salads, a 3-rd view video benchmark. Using the ViT backbone same as AVT, DCR achieves a 3.1% performance gain on Top-1 accuracy score. It shows DCR is a general training scheme not limited to egocentric action anticipation, but also advances anticipation performance in third-person videos.

4.5 Results of Early Action Recognition

We conduct experiments on Early Action Recognition problem on EK55 and EG+ validation sets. Results are reported in Tables 7 and 8, including Top-1 accuracy at different observation rates and Avg/Std/APD score described in Sect. 4.1.

On EK55, we use TSN backbones to encode RGB and Optical Flow feature of frames, and FRCNN to encode OBJ feature, same to the setting used in RULSTM (Furnari & Farinella , 2020). For the separated model on each modality, both the LSTM and default Transformer version of DCR perform better than the competitive RULSTM baseline. For

Table 4 Anticipation result ensemble on EK55

Method	Validation		Test Seen (S1)		Test Unseen (S2)	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
ATSN	–	16.3	6.0	28.2	2.3	9.4
ED	–	25.8	8.1	18.2	2.4	6.6
MCE	–	26.1	10.8	25.3	5.6	15.7
RULTSM	15.3	35.3	14.4	33.7	8.2	21.1
FHOI	10.4	25.5	15.4	34.3	8.6	22.9
ImagineRNN	–	35.6	14.7	35.0	9.3	22.2
ActionBanks	15.1	35.6	16.7	36.1	10.0	23.4
Ego-OMG	**19.2**	–	16.0	34.5	**11.8**	23.8
AVT	16.6	37.6	16.8	36.5	10.4	24.3
DCR	**19.2**	**41.2**	**17.7**	**38.5**	10.9	**24.8**

Numbers in bold are to highlight best performances under their settings

Table 6 Anticipation results on 50S

Method	Top-1
DMR	6.2
RNN	30.1
CNN	29.8
ActionBanks	40.7
AVT	48.0
DCR	**51.1**

Numbers in bold are to highlight best performances under their settings

example, using RGB feature and LSTM structure, DCR training strategy leads to 0.9% average accuracy gain, especially 1.2% gain at the lowest 12.5% observation rate. Our LSTM model is more robust than classic RULSTM as it has better performance measured by Std and APD. The more powerful Transformer model makes a further improvement, with 1.7% average accuracy gain and 0.39%/0.99% decreases on Std/APD over RULSTM. The results on the FLOW modal are similar. LSTM version DCR outperforms RULSTM with a 0.7% gain on Avg and 0.46%/0.37% decreases on Std/APD. The Transformer model in DCR training scheme does better, with 1.4% gain on Avg and 0.62%/0.61% decreases on Std/APD. On OBJ modal, LSTM version DCR also outperforms RULSTM. But we observe an interesting phenomenon between LSTM and Transformer. Though the powerful Transformer model achieves better accuracy average, its prediction ability seems to be not stable in this modality. LSTM

Table 7 Results of Early Action Recognition on EPIC-KITCHENS-55

Method	Top-1 action accuracy at different observation rates								Avg	Std ↓	APD ↓
	12.5%	25%	37.5%	50%	62.5%	75%	87.5%	100%			
RGB											
RULSTM	20.1	23.0	24.4	25.5	26.7	27.5	27.8	28.3	25.4	2.63	2.89
DCR(LSTM)	21.3	24.4	25.4	26.6	27.5	27.8	28.5	28.7	26.3	2.34	2.44
DCR	**22.2**	**25.3**	**26.0**	**27.6**	**28.5**	**28.9**	**28.9**	**29.0**	**27.1**	**2.24**	**1.90**
FLOW											
RULSTM	10.5	13.5	15.9	18.6	20.1	21.4	21.6	21.9	17.9	3.96	3.96
DCR(LSTM)	12.1	14.7	16.7	19.2	20.4	21.5	22.1	22.2	18.6	3.50	3.59
DCR	**13.0**	**15.8**	**17.4**	**19.8**	**21.1**	**22.3**	**22.7**	**22.5**	**19.3**	**3.34**	**3.35**
OBJ											
RULSTM	12.9	12.9	13.6	14.1	14.9	15.2	15.8	16.1	14.4	1.17	1.66
DCR(LSTM)	**14.2**	**15.4**	16.0	16.6	17.0	17.4	17.5	17.6	16.5	**1.11**	**1.09**
DCR	13.5	15.1	**16.4**	**17.3**	**18.3**	**18.3**	**18.7**	**19.1**	**17.1**	1.83	2.01
FUSION											
FN	19.6	23.9	25.7	26.9	27.5	28.3	28.2	28.4	26.1	2.84	**2.34**
RL	22.5	25.0	27.2	28.6	29.6	30.1	30.5	30.5	28.0	2.73	2.50
EL	19.7	23.3	26.0	27.5	29.1	29.9	30.9	31.4	27.2	3.79	4.17
LSTM	22.1	25.7	27.8	28.9	29.8	31.1	31.2	30.9	28.4	2.98	2.76
RULSTM	24.5	27.6	29.4	30.9	32.2	33.1	33.6	**34.1**	30.7	3.12	3.43
Ego-Omg	26.0	28.3	–	31.1	–	31.2	–	–	–	–	–
DCR	**26.1**	**29.6**	**30.8**	**32.5**	**33.4**	**33.7**	**34.1**	**34.1**	**31.8**	**2.66**	2.36

Numbers in bold are to highlight best performances under their settings

Table 8 Results of Early Action Recognition on EGTEA GAZE+

Method	Top-1 action accuracy at different observation rates								Avg	Std ↓	APD ↓
	12.5%	25%	37.5%	50%	62.5%	75%	87.5%	100%			
FN	44.0	50.3	53.3	55.1	56.6	57.3	57.9	57.7	54.0	4.50	3.88
RL	45.4	51.0	54.2	56.6	58.1	58.9	59.3	59.5	55.4	4.66	4.12
EL	40.3	48.1	51.8	54.7	56.9	58.5	59.6	**60.2**	55.7	6.38	6.44
LSTM	50.2	53.8	55.7	57.2	58.0	58.8	59.1	59.3	56.5	2.96	2.79
RULSTM	45.9	51.8	54.4	57.1	58.2	59.3	60.1	**60.2**	55.9	4.66	4.33
DCR	**51.5**	**54.8**	**57.2**	**58.5**	**59.4**	**60.0**	**60.3**	**60.2**	**57.7**	**2.93**	**2.57**

Numbers in bold are to highlight best performances under their settings

model does better at low-level observation rates and on the measurements of model robustness.

Same to the anticipation experiments, we also make model ensemble via late fusion. Our training scheme performs state-of-the-art methods at all levels of observation rates and achieves 1.1% average performance gain. Besides, our training strategy shows superiority in model robustness. Compared to RULSTM, DCR has the same recognition accuracy at full observation but 1.6% better at 12.5% observation rate. The performance drop is 8% for DCR, much better compared to 9.6% for RULSTM. DCR achieves 0.46%/1.07% decreases on Std/APD measurements, compared to RULSTM. DCR performs best on all metrics except APD. This is mainly because baseline FN generally shows poor ability in Early Action Recognition especially the accuracy at full observation.

Early Action Recognition results on EGTEA GAZE+ show in Table 8. We fuse the predicted results of RGB and FLOW input models to give final predictions. DCR also has approaching performance at full observation rates but does better at low-level observation, e.g. 5.6% performance gain over RULSTM at 12.5% observation rate. Generally, DCR achieves 1.2% average accuracy gain over the prior state-of-the-art methods and also turns out to be best on all robustness measurements.

4.6 Ablation Study

We conduct ablation studies to verify the effects of our method.

In Video Action Anticipation problem, we do experiments on EK100 and EG+ validation sets with RGB inputs. Tables 9 and 10 report results on Transformer and LSTM respectively. (1) First, we compare a classification baseline by removing every anticipation optimization. Each branch suffers a large performance drop, indicating basic classification technique is not suitable for direct anticipation. (2) Second, we analyze Transformer models without order-aware pre-train. Their performances degrade, especially a 2.4% drop on EG+. (3) Third, we analyze different training routes on easiness schedules. If we always train under T_e=1, it turns out that training and testing tasks have a large gap and models can't transfer well, especially for Transformer. If we always train without using future context as T_e=0, then the model gets trapped in a local optimum and performs not very well. We consider different global schedules of T_e, like linearly decreasing from 1 to 0 or exponentially multiplying $\gamma = 0.95$ after each epoch. Though these methods may keep competitive in some scenarios, they are empirically worse than our proposed training route. (4) Last, we validate the effects of loss components. Our model turns out to have the largest performance drop without L_{rec}, even worse than the classification. This is because different context complicates classification without feature-level supervision. Moreover, without label smoothing, we observe a quick loss decrease in training and worse performance due to overfitting.

In Early Action Recognition problem, we do experiments on EK55 validation set with RGB inputs. Results are reported in Table 11. (1) First, we remove order-aware pre-training for Transformers. Model performance degrades especially at low-level observation, e.g., 1.0% drop at 25% observation. (2) Second, we consider different schemes in sampling multiple training tasks. An alternative is to use vanilla uniform distribution in sampling. But without balancing different training difficulties of tasks, model performances hardly live up to the standard DCR, especially on Std and APD measurements. We also try to directly apply single-task training route in Early Action Recognition. Though it may get stuck in certain tasks of more observations and perform better on easier tasks, its overall performance hardly reaches expectations. Both Transformer and LSTM models have large performance drops at 12.5% observation rates and turn out to be less robust. (3) Last, we validate the effects of loss components. Removing L_{rec} leads to 0.8% and 1.3% average accuracy drops for

Table 9 Ablation study of anticipation on Transformer

	EK100		EG+
	TSM	TSN	TSN
DCR	**16.1**	**14.6**	**64.5**
classification	13.7	12.7	58.5
w.o. pre-training	15.5	14.3	62.1
$T_e = 1$	6.5	4.5	40.1
$T_e = 0$	15.2	13.8	62.9
linear T_e	15.0	13.9	64.0
exponential T_e	15.6	14.2	64.2
w.o. L_{rec}	13.5	12.6	56.0
w.o. label smooth	14.8	13.3	62.3

Numbers in bold are to highlight best performances under their settings

Table 10 Ablation study of anticipation on LSTM

	EK100	
	TSM	TSN
DCR(LSTM)	**15.2**	**14.5**
classification	14.0	13.5
$T_e = 1$	14.1	13.1
$T_e = 0$	14.6	13.9
linear T_e	15.0	14.2
exponential T_e	**15.2**	14.4
w.o. L_{rec}	14.5	13.8
w.o. label smooth	14.0	13.3

Numbers in bold are to highlight best performances under their settings

Table 11 Ablation study of Early Action Recognition on EK55 with RGB-TSN backbone

Method	Top-1 action accuracy at different observation rates								Avg	Std ↓	APD ↓
	12.5%	25%	37.5%	50%	62.5%	75%	87.5%	100%			
DCR	**22.2**	**25.3**	**26.0**	**27.6**	**28.5**	**28.9**	28.9	29.0	**27.1**	2.24	**1.90**
w.o. pre-training	21.9	24.3	25.8	27.5	28.3	**28.9**	28.8	28.6	26.8	2.38	2.13
uniform sampling	21.8	24.7	26.0	27.2	28.4	28.6	28.9	29.0	26.8	2.38	2.17
single task route	19.4	23.6	25.3	27.4	28.3	**28.9**	**29.7**	**29.9**	26.5	3.38	3.35
w.o. L_{rec}	21.6	24.7	25.4	26.7	27.2	27.8	28.2	28.5	26.3	**2.12**	2.22
w.o. L_{con}	21.4	25.1	26.0	27.5	28.1	28.7	29.0	28.5	26.8	2.40	2.19
w.o. label smooth	20.9	24.0	25.6	26.5	27.8	28.2	28.5	28.3	26.2	2.47	2.26
DCR (LSTM)	**21.3**	**24.4**	**25.4**	**26.6**	**27.5**	**27.8**	**28.5**	**28.7**	**26.3**	**2.34**	**2.44**
uniform sampling	20.4	22.9	24.2	25.5	26.5	27.3	28.2	28.2	25.4	2.58	2.80
single task route	17.3	21.6	24.4	25.2	26.5	27.3	27.8	27.2	24.7	3.35	3.12
w.o. L_{rec}	19.7	22.7	24.1	25.2	26.0	27.1	27.6	27.7	25.0	2.58	2.72
w.o. L_{con}	20.5	23.2	25.2	26.5	**27.5**	**27.8**	28.2	28.6	25.9	2.64	2.68
w.o. label smooth	21.2	23.9	25.2	26.3	27.4	**27.8**	**28.5**	28.5	26.1	2.40	**2.44**

Numbers in bold are to highlight best performances under their settings

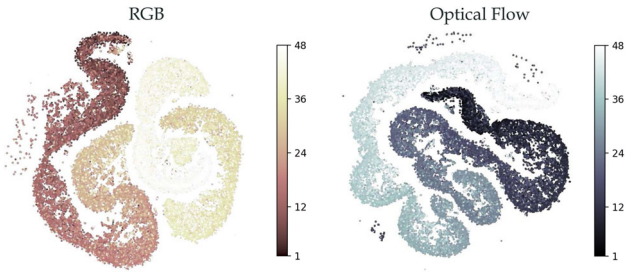


Fig. 7 Effect of order-aware pre-training. We sample 1000 segments from EK100 validation set and color the order-aware tokens from pre-training models according to their temporal positions. The models have learned temporal dynamics (Color figure online)

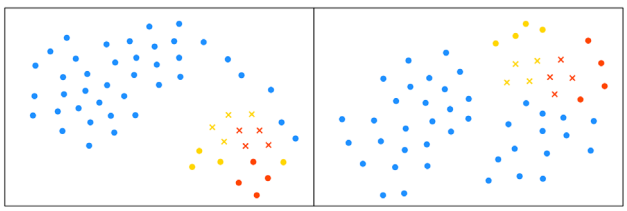


Fig. 8 Qualitative cases of frame reconstruction. Blue dots are visible frames. Crosses are the reconstructed future representation, much closer to the exact frames (yellow and red dots) (Color figure online)

Transformer and LSTM. Though the average accuracy drop for removing L_{con} is marginal 0.3%, L_{con} affects more at low observation rate (e.g., 0.8% at 12.5% observation) and benefits robustness metrics (Std, APD).

4.7 Qualitative Results

We give qualitative results to better characterize the reasoning ability of our method.

First, we show what the model learns in the order-aware pre-training phase. We sample 1000 segments from EK100 validation set and extract output tokens from pre-trained order-aware Transformer. In Fig. 7, we use t-SNE (Van der Maaten & Hinton , 2008) to embed them into a 2D space and color frames according to their temporal positions. It shows our pre-trained models have learned video dynamics in the latent manifold, with a more comprehensive understanding of temporal logic in videos.

Next, we show qualitative cases of frame reconstruction. All frames and the model reconstruction are embedded via t-SNE and scattered in Fig. 8. Blue ones are observed in the anticipation task, while the yellow and red ones are future frames. The reconstructed frames marked as crosses are closer to the cluster of future frames.

5 Conclusion

In this paper, we propose a novel strategy DCR on how to train models to tackle video action predictive problems including Video Action Anticipation and Early Action Recognition. We design an effective curriculum training route for the single-task Video Action Anticipation. Besides, we observe the multi-tasking nature of Early Action Recognition and propose new techniques on task sampling and learning constraints. Our training strategy follows the intuitive learning process of humans and flexibly advances widely-used reasoning models in effectiveness and efficiency. In extensive experiments, we establish new *state-of-the-art* results on several widely used Video Action Anticipation and Early Action Recognition benchmarks.

However, there are also some limitations and potential negative impacts on our work. Our work is empirically validated but more theories need to be done in future works. We train models on human-annotated datasets, which may import bias from human-defined labels. A possible further solution for debiasing is to utilize unsupervised learning techniques on a larger scale of data. On the usage of our method, action prediction techniques are generally harmless except for some malicious use for bad-intended prediction. We encourage a proper use of technology that benefits mankind. [1]

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11263-023-01850-6.

Acknowledgements We appreciate the support from National Natural Science Foundation of China (No.72192821, 72192820), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), Shanghai Qi Zhi Institute, and SHEITC (2018-RGZN-02046).

Data Availability Our codes, models and pre-extracted features are publicly available at https://github.com/AllenXuuu/DCR. The raw data can be downloaded from the public websites of datasets.

References

Alvarez, W. M., Moreno, F. M., Sipele, O., Smirnov, N., & Olaverri-Monreal, C. (2020). Autonomous driving: Framework for pedestrian intention estimation in a real world scenario. In *2020 IEEE intelligent vehicles symposium (IV)* (pp. 39–44). IEEE.

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., & Schmid, C. (2021). Vivit: A video vision transformer. Preprint retrieved from arXiv:2103.15691

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning* (pp. 41–48).

Camporese, G., Coscia, P., Furnari, A., Farinella, G. M., & Ballan, L. (2021). Knowledge distillation for action anticipation via label smoothing. In *2020 25th international conference on pattern recognition (ICPR)* (pp. 3312–3319). IEEE.

Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? A new model and the kinetics dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

[1] EPIC-KITCHENS: https://epic-kitchens.github.io EGTEA GAZE+: https://cbs.ic.gatech.edu/fpv/#egtea_gaze_plus 50-Salads: https://cvip.computing.dundee.ac.uk/datasets/foodpreparation/50salads/

Cirik, V., Hovy, E., Morency, & L. P. (2016). Visualizing and understanding curriculum learning for long short-term memory networks. Preprint retrieved from arXiv:1611.06204

Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., & Wray, M. (2018). Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 720–736).

Damen, D., Doughty, H., Farinella, G.M., Furnari, A., Kazakos, E., Ma, J., Moltisanti, D., Munro, J., Perrett, T., Price, W., & Wray, M. (2021). Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. International Journal of Computer Vision (IJCV). https://doi.org/10.1007/s11263-021-01531-2.

De Geest, R., & Tuytelaars, T. (2018). Modeling temporal structure with LSTM for online action detection. In *2018 IEEE winter conference on applications of computer vision (WACV)* (pp. 1549–1557). IEEE.

Dessalene, E., Devaraj, C., Maynord, M., Fermuller, C., & Aloimonos, Y. (2021). Forecasting action through contact representations from first person video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* p. 1. https://doi.org/10.1109/tpami.2021.3055233

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., & Solorio, T. (Eds.) *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies*, NAACL-HLT 2019, Minneapolis, Vol. 1(Long and Short Papers). Association for Computational Linguistics (pp. 4171–4186). https://doi.org/10.18653/v1/n19-1423

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. & Uszkoreit, J. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. Preprint retrieved from arXiv:2010.11929

Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., & Feichtenhofer, C. (2021). Multiscale vision transformers. Preprint retrieved from arXiv:2104.11227

Farha, Y. A., Richard, A., & Gall, J. (2018). When will you do what?—Anticipating temporal occurrences of activities. Preprint retrieved from arXiv:1804.00892

Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2019). Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6202–6211).

Foo, L. G., Li, T., Rahmani, H., Ke, Q., & Liu, J. (2022). Era: Expert retrieval and assembly for early action prediction. In *Computer vision–ECCV 2022: 17th European conference*, Tel Aviv, Proceedings, Part XXXIV , (pp. 670–688). Springer.

Furnari, A., Battiato, S., Grauman, K., et al. (2017). Next-active-object prediction from egocentric videos. *Journal of Visual Communication and Image Representation, 49*, 401–411. https://doi.org/10.1016/j.jvcir.2017.10.004

Furnari, A., Battiato, S., & Maria Farinella, G. (2018). Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation. In *Proceedings of the European conference on computer vision (ECCV) workshops*.

Furnari, A., & Farinella, G. (2020). Rolling-unrolling LSTMS for action anticipation from first-person video. IEEE transactions on pattern analysis and machine intelligence.

Gao, J., Yang, Z., & Nevatia, R. (2017). Red: Reinforced encoder-decoder networks for action anticipation. Preprint retrieved from arXiv:1707.04818

Girdhar, R., & Grauman, K. (2021). Anticipative Video Transformer. In *ICCV*

Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., & Hoppe, F. (2017). The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision* (pp. 5842–5850).

Gu, X., Qiu, J., Guo, Y., Lo, B., & Yang, G. Z. (2021). Transaction: Icl-sjtu submission to epic-kitchens action anticipation challenge 2021. Preprint retrieved from arXiv:2107.13259

Hayat, M., Khan, S., Zamir, S. W., Shen, J., & Shao, L. (2019). Gaussian affinity for max-margin class imbalanced learning. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

Hu, J. F., Zheng, W. S., Ma, L., et al. (2019). Early action prediction by soft regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 41*(11), 2568–2583. https://doi.org/10.1109/TPAMI.2018.2863279

Jain, A., Singh, A., Koppula, H. S., Soh, S., & Saxena, A. (2016). Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In *2016 IEEE international conference on robotics and automation (ICRA)* (pp. 3118–3125). IEEE.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., & Suleyman, M. (2017). The kinetics human action video dataset. Preprint retrieved from arXiv:1705.06950

Ke, Q., Fritz, M., & Schiele, B. (2019). Time-conditioned action anticipation in one shot. In *CVPR*

Kong, Y., Tao, Z., & Fu, Y. (2017). Deep sequential context networks for action prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

Koppula, H. S., & Saxena, A. (2015). Anticipating human activities using object affordances for reactive robotic response. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(1), 14–29.

Kuehne, H., Arslan, A., & Serre, T. (2014). The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *2014 IEEE conference on computer vision and pattern recognition* (pp. 780–787). https://doi.org/10.1109/CVPR.2014.105

Kumar, M., Packer, B., & Koller, D. (2010). Self-paced learning for latent variable models. *Advances in Neural Information Processing Systems, 23*, 1189–1197.

Li, S., Zhu, X., Huang, Q., Xu, H., & Kuo, C. C. J. (2017). Multiple instance curriculum learning for weakly supervised object detection. Preprint retrieved from arXiv:1711.09191

Li, Y., Liu, M., & Rehg, J. M. (2018). In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European conference on computer vision (ECCV)*

Lin, J., Gan, C., & Han, S. (2019). TSM: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*

Liu, M., Tang, S., Li, Y., & Rehg, J. M. (2020). Forecasting human-object interaction: joint prediction of motor attention and actions in first person video. In *European conference on computer vision* (pp. 704–721). Springer.

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. Preprint retrieved from arXiv:1711.05101

Ma, S., Sigal, L., & Sclaroff, S. (2016). Learning activity progression in LSTMs for activity detection and early detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1942–1950).

Misra, I., Zitnick, C. L., & Hebert, M., et al. (2016). Shuffle and learn: Unsupervised learning using temporal order verification. In B. Leibe, J. Matas, & N. Sebe (Eds.), *Computer Vision—ECCV 2016* (pp. 527–544). Cham: Springer International Publishing.

Pang, G., Wang, X., Hu, J., Zhang, Q., & Zheng, W. S. (2019). Dbdnet: Learning bi-directional dynamics for early action prediction. In: IJCAI (pp. 897–903). https://doi.org/10.24963/ijcai.2019/126

Pang, Y., Ke, Q., Rahmani, H., Bailey, J., & Liu, J. (2022). Igformer: Interaction graph transformer for skeleton-based human interaction recognition. In Part, X. X. V. (Ed.), *Computer Vision-ECCV 2022: 17th European Conference* Tel Aviv (pp. 605–622). Springer.

Rasouli, A., Kotseruba, I., & Tsotsos, J. K. (2019). Pedestrian action anticipation using contextual feature fusion in stacked RNNs. In *BMVC*.

Ryoo, M. S. (2011). Human activity prediction: Early recognition of ongoing activities from streaming videos. In *2011 international conference on computer vision* (pp. 1036–1043). IEEE.

Ryoo, M. S., Fuchs, T. J., Xia, L., Aggarwal, J. K., & Matthies, L. (2015). Robot-centric activity prediction from first-person videos: What will they do to me? In *2015 10th ACM/IEEE international conference on human-robot interaction (HRI)* (pp. 295–302).

Sener, F., Singhania, D., & Yao, A. (2020). Temporal aggregate representations for long-range video understanding. Preprint retrieved from arXiv:2006.00830

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. Advances in Neural Information Processing Systems *1*.

Srivastava, N., Mansimov, E., & Salakhudinov, R. (2015). Unsupervised learning of video representations using LSTMs. In *International conference on machine learning* (pp. 843–852). PMLR.

Stein, S., & McKenna, S. J. (2013). Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing* (pp. 729–738).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818–2826).

Tran, D., Wang, H., Torresani, L., & Feiszli, M. (2019). Video classification with channel-separated convolutional networks. In *2019 IEEE/CVF international conference on computer vision (ICCV)* (pp. 5551–5560). https://doi.org/10.1109/ICCV.2019.00565

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., & Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *CVPR* (pp. 6450–6459).

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*(11).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł, & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Vondrick, C., Pirsiavash, H., & Torralba, A. (2016). Anticipating visual representations from unlabeled video. In *CVPR*.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., & Van Gool, L. (2016). Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision* (pp. 20–36). Springer.

Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7794–7803).

Wang, X., Hu, J. F., Lai, J. H., Zhang, J., & Zheng, W. S. (2019). Progressive teacher-student learning for early action prediction. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 3551–3560). https://doi.org/10.1109/CVPR.2019.00367

Weinshall, D., Cohen, G., & Amir, D. (2018). Curriculum learning by transfer learning: Theory and experiments with deep networks. In

International conference on machine learning (pp. 5238–5246). PMLR.

Wu, C. Y., Feichtenhofer, C., Fan, H., He, K., Krahenbuhl, P., & Girshick, R. (2019). Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

Wu, Y., Zhu, L., Wang, X., et al. (2021). Learning to anticipate egocentric actions by imagination. *IEEE Transactions on Image Processing, 30*, 1143–1152. https://doi.org/10.1109/tip.2020.3040521

Xu, X., Li, Y. L., & Lu, C. (2022). Learning to anticipate future with dynamic context removal. In *CVPR*.

Zhang, Y., Abbeel, P., & Pinto, L. (2020). Automatic curriculum learning through value disagreement. Advances in Neural Information Processing Systems *33*

Zhou, Y., & Berg, T. L. (2015). Temporal perception and prediction in ego-centric video. In *Proceedings of the IEEE international conference on computer vision* (pp. 4498–4506).