AdamsFormer for Spatial Action Localization in the Future

Hyung-gun Chi[† 2] Kwonjoon Lee[1] Nakul Agarwal[1] Yi Xu[1,3] Karthik Ramani[2] Chiho Choi[† 4]

[1]Honda Research Institute USA [2]Purdue University [3]Northeastern University [4]Samsung Semiconductor US

{hgchi, ramani}@purdue.edu {kwonjoon_lee, nakul_agarwal}@honda-ri.com

xu.yi@northeastern.edu chiho1.choi@samsung.com

Abstract

Predicting future action locations is vital for applications like human-robot collaboration. While some computer vision tasks have made progress in predicting human actions, accurately localizing these actions in future frames remains an area with room for improvement. We introduce a new task called spatial action localization in the future (SALF), which aims to predict action locations in both observed and future frames. SALF is challenging because it requires understanding the underlying physics of video observations to predict future action locations accurately. To address SALF, we use the concept of NeuralODE, which models the latent dynamics of sequential data by solving ordinary differential equations (ODE) with neural networks. We propose a novel architecture, AdamsFormer, which extends observed frame features to future time horizons by modeling continuous temporal dynamics through ODE solving. Specifically, we employ the Adams method, a multi-step approach that efficiently uses information from previous steps without discarding it. Our extensive experiments on UCF101-24 and JHMDB-21 datasets demonstrate that our proposed model outperforms existing long-range temporal modeling methods by a significant margin in terms of frame-mAP.

1. Introduction

Human action understanding is essential in computer vision, especially for applications like VR/AR [61, 64], robotics [57, 60], and autonomous vehicles [28, 38]. These applications help users by interpreting intentions or perceiving others' actions in the environment. Considerable progress has been made in human action perception, including action recognition [7, 10, 11], temporal action localization [3, 36], and spatio-temporal action localization [2, 30, 42, 52].

Lately, predicting and anticipating human actions, such as early action prediction [13, 23, 62], action anticipation [15, 16], and hand or pedestrian trajectory prediction [40, 45, 47], have gained attention due to the increasing need to prepare for future events. While progress has been made in predicting human actions, further exploration is needed in localizing future actions, which is critical for various applications. For example, anticipatory behavior is essential for effective collaboration in human-robot interaction [57]. Accurately predicting future activity locations enables robot agents to support humans more efficiently.

Figure 1. Future Spatial Action Localization (SALF) aims to identify diverse action patterns in both observed and future frames. Green and red boxes represent observed and future frames, while blue bounding boxes indicate predicted action locations.

In this work, we introduce Spatial Action Localization in the Future (SALF), a novel task that expands upon traditional spatio-temporal action localization. SALF aims to predict spatial locations and categorize actions in both long-term future and past observations, as illustrated in Fig. 1. By enabling models to recognize and classify present actions while anticipating and localizing future actions, SALF significantly enhances real-time decision-making and adaptive responses in complex environments. As demonstrated in Table 1, SALF uniquely focuses on diverse, highly nonlinear motion patterns across various action categories, setting it apart from related tasks like pedestrian or hand trajectory prediction. Additionally, SALF differs in input and target, utilizing only video input and predicting multiple bounding boxes and action categories for both future and observed frames. In contrast, trajectory predictions generally estimate the future path of specific objects, such as hands or pedestrians, without requiring bounding boxes.

To address the challenges of SALF, we leverage the con-

[†] Work done while at Honda Research Institute USA.

Task	Input		Target	
	Video	BBox	Trajectory	Classification
Pedestrian prediction [47]	✓	✓	Bbox (F)	Intention (Binary)
Pedestrian prediction [45]		✓	Bbox (F)	-
Hand prediction [40]	✓	✓	Center (F)	-
SALF (ours)	✓		Bbox (O & F)	Actions

Table 1. Comparison between SALF and trajectory predictions. 'Bbox' signifies the bounding box, while 'O' and 'F' represent observation and future, respectively

cept of NeuralODE [6]. Recent works on Neural ODE [6,49,67] and its applications [26,27,35,43,65] demonstrate that Neural ODE successfully models continuous sequential data by solving ordinary differential equations (ODE) with neural networks. Neural ODE has an advantage over other temporal modeling methods like transformers [58] or RNNs [21] in that it can model the underlying physics of sequential data. We adapt the concept of Neural ODE to predict information for future frames from observations to address the proposed SALF.

From this motivation, We propose *AdamsFormer*, a network designed to detect spatial locations of the action for *both* the *observed* previous frames and *unobserved* future frames. The proposed model predicts future action locations by extrapolating observed frames' latent features to the future time horizon we want to predict. With the extrapolated latent features of future frames, we can predict the locations of the action and their corresponding categories. When solving ODE, we adopt the multi-step method (Adams method), which is more robust to noisy conditions than single-step methods such as Euler or Runge-Kutta. A single-step method that uses information from only the previous step can be easily affected by noise. In contrast, a multi-step way attends several previous steps to predict the future; thus, it gains efficiency and robustness by using the information from previous frames rather than discarding it. Using a toy example, we compare multi-step and single-step methods in Fig. 2.

We conduct extensive experiments on action video datasets UCF101-24 [54] and JHMDB-21 [32] to demonstrate the advantage of the proposed architecture and benchmark the existing long-range temporal dependency modeling algorithms on SALF. We observe that *AdamsFormer* outperform other state-of-the-art models, thus demonstrating its efficacy. We also provide a deeper analysis to provide intuition to researchers on how to improve the model performance on SALF.

In summary, our contributions are as follows,

- We present a novel task called Spatial Action Localization in the Future (SALF), which aims to identify the spatial boundaries of actions in both observed and future frames.
- To address the SALF task, we introduce AdamsFormer, an innovative architecture that predicts action locations in future frames by extrapolating the latent state using the Adams method to solve ODEs.
- Our extensive experimental results demonstrate that AdamsFormer significantly outperforms existing state-of-the-art methods for long-range feature modeling in the SALF task.

2. Related Work

Spatio-Temporal Action Localization This task aims to localize atomic actions in videos with 3D spatio-temporal bounding boxes. Motivated by object detection methods, previous works [2, 30, 44, 52, 63] localize action in frame-level and link frame-wise predictions to make temporal bounds. Some work [17, 55] apply 3D CNN to capture temporal information, and others [12,42,56] focus on modeling the relation between captured actors or object-actor to classify the actions better. Another branch of this task is tubelet-level action detection [33, 34, 53, 68], which was first introduced by [25]. Hou *et al.* [22] first proposed a 3D cuboid proposal method, and Yang *et al.* [66] introduced a method that progressively refined the proposals. More recently, Zhao *et al.* [69] introduced a transformer [58]-based method that can detect action tubelet with queries as proposals. Our task is also localizing Spatial bounding boxes of action on the observation, but, different from these works, we are more focused on localizing activity for future frames.

Long-term Action Anticipation Action anticipation is the task that predicts a sequence of actions in the future based on the partial observation of past actions. Various approaches have been proposed to tackle this task in both third person view [14, 24, 31, 59] and egocentric view [8, 39, 41] actions. The recent works [15, 16] utilize transformer [58] to anticipate action in the future, leveraging the advantage of self-attention learning temporal interactions. Recently, long-term action anticipation [1, 16, 29, 50] gained popularity as the need to predict the distant future grows in many computer vision applications. Unlike long-term action anticipation, SALF aims to predict the location of the future action and the category that makes the task more challenging than action anticipation.

Video Prediction Video prediction [9,18,43] is a task that predicts future frames given past frames. Despite significant community efforts in this domain, faithful modeling of long-term future frames remains an area for improvement. Compared to video prediction, the newly introduced SALF makes action dynamics modeling much easier by abstracting away appearance variations. It may be more beneficial for human-robot interaction since it provides machines with more direct access to future locations of human action.

3. Spatial Action Localization in the Future

We introduce a novel task, Spatial Action Localization in the Future (SALF), which aims to predict the location and

corresponding categories of activities in each frame of a future video sequence. Formally, let $\mathbf{X}_{1:t} = \{\mathbf{X}_1, \cdots, \mathbf{X}_t\}$ represent a video with t frames. Given the first O frames of the video, $\mathbf{X}_{1:O}$, the task involves localizing and classifying actions in both observed and future T frames. This challenge is particularly difficult because localizing future actions demands more information than early action prediction, which only predicts the action class category.

4. Proposed Method

4.1. Background

Initial Value Problem (IVP) and NeuralODE An initial value problem (IVP) is an instance of an Ordinary Differential Equation (ODE) with an initial condition specifying the value of the unknown function at a point. Formally speaking, an IVP can be expressed as follows:

$$z'(t) = \frac{dz}{dt} = f(t, z), z(t_0) = z_0, \tag{1}$$

where the function $z(t)$ is a solution of IVP. Recently, Neural ODE [6, 49] models ODE function $f(\cdot)$, a derivative of $z(t)$, with neural network for continuous times-series modeling. Following these works, we also use the concept of IVP with Neural ODE to model the dynamics of video.

Numerical methods for IVP Obtaining an analytical solution to a differential equation is often infeasible. In this case, we must resort to approximate methods that numerically approximate the integration of derivatives with a finite sum. Single-step methods such as Euler's method consider only the derivative at one previous step to determine the function value at the current step. Concretely, given the step size of h,

$$z_{n+1} = z_n + hf(t_n, z_n). \tag{2}$$

Multi-step methods improve the precision of numerical approximation by considering function values from previous $N(\geq 2)$ steps. For example, second-order ($N = 2$) and third-order ($N = 3$) Adams-Bashforth method can be expressed as:

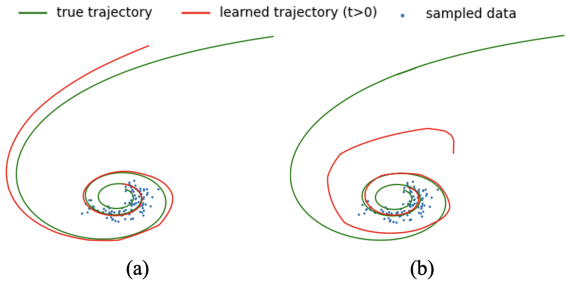
$$(N=2): z_{n+1} = z_n + h[\frac{3}{2}f(t_n, z_n) - \frac{1}{2}f(t_{n-1}, z_{n-1})],$$
$$(N=3): z_{n+1} = z_n + h[\frac{23}{12}f(t_n, z_n) - \frac{16}{12}f(t_{n-1}, z_{n-1}) + \frac{5}{12}f(t_{n-2}, z_{n-2})].$$

A step-size h is set to 1 in our work.

4.2. Problem Formulation

We formulate Spatial Action Localization in the Future (SALF) task as extrapolating latent features of observed frames. More formally, the task is predicting the latent feature of future frames $\mathbf{Z}_{O+1:O+T} = \{\mathbf{Z}_{O+1}, \cdots, \mathbf{Z}_{O+T}\}$, by extrapolating those of initial observation $\mathbf{Z}_{1:O} = \{\mathbf{Z}_1, \cdots, \mathbf{Z}_O\}$, where $\mathbf{Z}_t$ is a latent feature of the frame at time t. The symbol T and O denote the video prediction length and observation length, respectively. For this, we convert the SALF task as the Initial Value Problem with an initial condition as a latent feature of the last observed frame $z(0) = \mathbf{Z}_O$ as follows,

$$z(t) = z(0) + \int_0^t f_\theta(\tau, z(\tau))d\tau. \tag{3}$$



(a) (b)

Figure 2. Toy example results of (a) multi-step (b) single-step. We can see that the multi-step method better captures underlying physics compared to the single-step way.

4.3. AdamsFormer

To tackle SALF, we propose a novel architecture that can solve the initial value problem (IVP) defined at Eq. (3) with the numerical method. We adapt the linear multi-step method to solve IVP, which leverages previous steps to calculate the next value, whereas the single-step way takes only one previous step. We selected the multi-step approach since it is known for being more robust to the stiff equation than the single-step method. Action videos often contain noise like camera motion to record the dynamic movement of action, which hampers accurate action prediction. A multi-step based approach can robustly predict the future by attending multiple previous steps, similar to the smoothing effect of sliding windows. We compare multi-step and single-step in Fig. 2 with a toy example to predict the trajectory of spiral function from sampled data from the trajectory, We see that the multi-step method better captures underlying physics (spiral function) than the single-step. For the linear multi-step method for ODE, we use the Adams method, one of the most popular methods, and named our proposed architecture *AdamsFormer* after the Adams method.

An overview of the proposed architecture is illustrated in Fig. 3 Left. The AdamsFormer follows three steps to solve SALF. First, the video encoder extracts latent features from video clips. Next, the future feature predictor extrapolates future features by solving IVP with the multi-step method. Lastly, the decoder localizes and classifies the future clip's action using extrapolated future features.

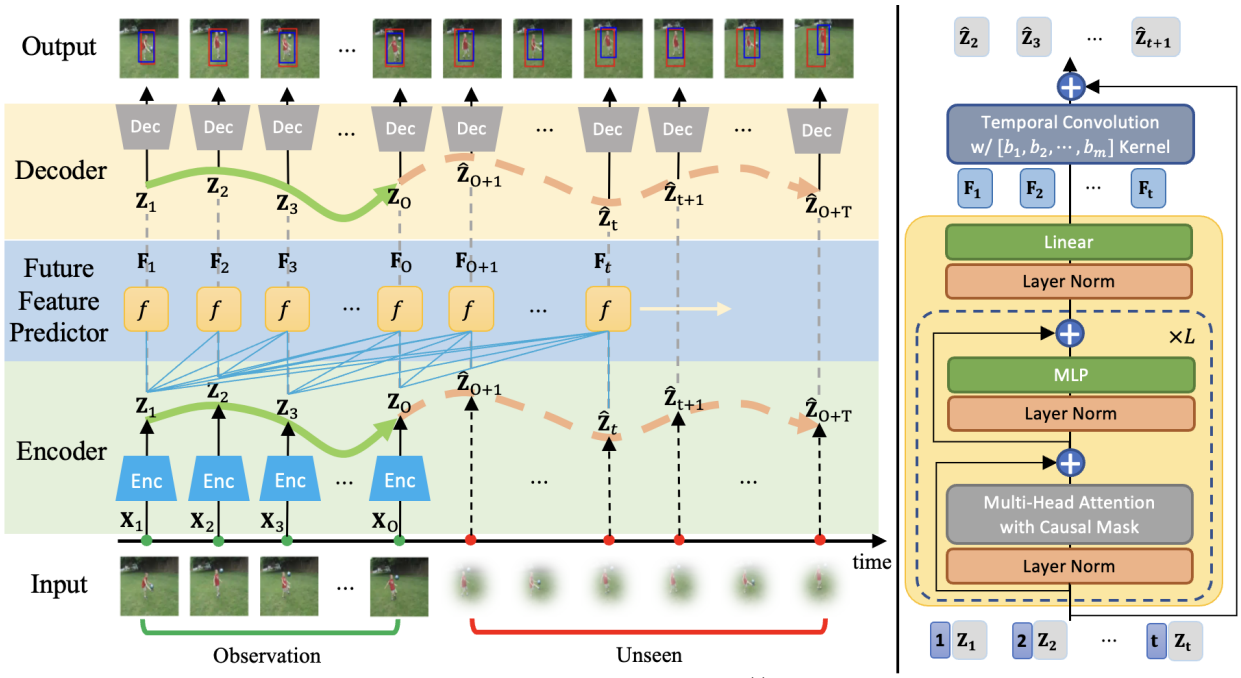


Figure 3. The overview of Adamsformer **(Left)** and detail about ODE function $f(\cdot)$ **(Right)**. The yellow arrow on the left figure indicates the ongoing direction of the sequence. Red and blue bounding boxes in the output indicate predictions and ground truth, respectively.

4.3.1 Video Encoder

The video encoder takes a video clip as input and produces a corresponding latent feature. To fully utilize temporal information, we combine features from 3D-CNN and 2D-CNN (i.e. DarkNet [48] and 3D-ResNet [19], respctively) following [30]. An input video clip at time t, $\mathbf{X}_t \in \mathbb{R}^{H\times W\times L\times 3}$, is passed through 3D-CNN and 2D-CNN and their outputs are concatenated together to construct latent feature $\mathbf{Z}_t \in \mathbb{R}^{H'\times W'\times D}$ for video clip. Here, H, W, and L denote height, width, and the number of frames in a video clip, respectively. For 2D-CNN, the last frame of video clip $\mathbf{X}_t$ is used as an input.

$$\mathbf{Z}_t = [\mathbf{Z}_t^{3D}||\mathbf{Z}_t^{2D}]\mathbf{W}, \tag{4}$$

where $\mathbf{W} \in \mathbb{R}^{(D^{2D}+D^{3D})\times D}$ is a learnable linear projection matrix and $[\cdot||\cdot]$ denotes concatenation of tensors among channel dimension. $\mathbf{Z}_t^{2D}$ and $\mathbf{Z}_t^{3D}$ represent output of 2D-CNN and 3D-CNN, respectively.

4.3.2 Future Feature Predictor

The next step is extrapolating the observed clips' latent features to reach for the future. As mentioned above, we solve Eq. (3) with a numerical method, precisely a linear multi-step method, formulated as follows.

$$\mathbf{Z}_{t+1} = \mathbf{Z}_t + h\sum_{j=0}^{N} b_j \mathbf{F}_{t-j}, \tag{5}$$

where $t \leq T$, $\mathbf{F}_i$ is the output of ODE function $f_\theta(t, \mathbf{Z}_i)$, and N is the number of step for multi-step method, and h is step-size which is set to 1 in our setup. Coefficients from N-th order Adams–Bashforth method, are used for b. The detailed design of the ODE function $f_\theta(t, \mathbf{Z}_i)$ will be explained in the following section. To implement Eq. (5), we use convolution along temporal axis of $\mathbf{F}_{1:t}$ using $\mathbf{b} = \{b_0, b_1, \ldots, b_{N-1}\}$ as $(N\times 1)$ kernel and add $\mathbf{Z}_t$ as a residual.

$$\mathbf{Z}_{t+1} = \mathbf{Z}_t + (\mathbf{F}_{1:t} * \mathbf{b})[t], \tag{6}$$

where symbol $*$ denotes a convolution operator. The implementation is depicted in Fig. 3 Right.

ODE Function We design an ODE function $f(\cdot)$ that models the dynamics of latent features using a Transformer [58] decoder with a causal mask. It takes all previous latent features $\mathbf{Z}_{1:i}$ rather than only current frame $\mathbf{Z}_i$. If $f(\cdot)$ only takes the latent feature of the current video clip $\mathbf{Z}_i$, it is only dependent on the current video clip ignoring all contexts of the video. For example, video clips with similar motion 'running' can be shown in many different action categories like *Fencing* and *Pole vault* in UCF24-21 [54] dataset. If the $f(\cdot)$ only takes $\mathbf{Z}_i$, $f(\cdot)$ will output a similar value despite the context of action. Therefore, to provide more contextual information of action for modeling dynamics of latent feature, we design $f(\cdot)$ to capture the context of the video by feeding latent feature of all previous frames $\mathbf{Z}_{1:i}$ as follows.

$$\mathbf{F}_i = f_\theta(\mathbf{t}_{1:i}, \mathbf{Z}_{1:i}), \tag{7}$$

where $\mathbf{t}_{1:i} = \{t_1, \cdots, t_i\}$ indicates a set of all previous times until i-th time.

We model $f(\cdot)$ with a neural network. An overview of the ODE function is illustrated in the yellow box of Fig. 3 Right. We use self-attention to let the model attend to latent features at different times. Sinusoidal positional embedding (PE$(\cdot)$) is added to each $\mathbf{Z}_t$ in different time t to construct initial hidden states $\mathbf{H}_t^{(0)} \in \mathbb{R}^{H' \times W' \times D}$.

$$\mathbf{H}_t^{(0)} = \mathbf{Z}_t + \text{PE}(t). \tag{8}$$

Here, PE$(\cdot) \in \mathbb{R}^{1\times1\times D}$ is broadcasted to the spatial dimension H' and W'. After the linear projection of $\mathbf{H}$, we derive Key $\mathbf{K}$, Query $\mathbf{Q}$, and Value $\mathbf{V}$ embeddings with the same size as $\mathbf{H}$ for Self-Attention. Then, we calculate temporal relations of the same spatial location (x, y) among embeddings with all different time horizons:

$$\begin{aligned}&\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V})[x, y] \\ &= \text{Softmax}_j(\frac{\mathbf{Q}_i[x, y]\mathbf{K}_j[x, y]^\top}{\sqrt{D}})\mathbf{V}_j[x, y],\end{aligned} \tag{9}$$

where i, j are temporal indices and x, y are spatial indices in the range of $1 \leq x \leq W'$ and $1 \leq x \leq H'$. We use multi-head attention (MSA) to explore subspaces of different representations of hidden states and apply the causal mask to guide the model to attend only to previous frames.

$$\text{MSA}(\mathbf{H}) = [\text{head}_1 || \cdots || \text{head}_k]\mathbf{W}', \tag{10}$$

$$\text{head}_i = \text{Attn}(\mathbf{Q}^i, \mathbf{K}^i, \mathbf{V}^i) \tag{11}$$

where $\mathbf{W}' \in \mathbb{R}^{D\times D}$ is a learnable weight matrix and k is the number of head. Layer-wise update rule of latent feature $\mathbf{H}$ is following,

$$\mathbf{H}^{(l+1)} = \text{LN}(\text{MLP}(\mathbf{H}'^{(l)})) + \mathbf{H}'^{(l)}, \tag{12}$$

$$\mathbf{H}'^{(l)} = \text{LN}(\text{MSA}(\mathbf{H}^{(l)})) + \mathbf{H}^{(l)}, \tag{13}$$

where LN$(\cdot)$ denotes layer norm [4]. Two fully-connected layers with GeLU [20] as an activation function are used for MLP$(\cdot)$. We stack L number of the above layers, and finally, $\mathbf{F}_t$ is derived by passing through the output of the last layer $\mathbf{H}^{(L)}$ to the linear layer.

4.3.3 Decoder

The decoder takes $\mathbf{Z}_t$, derived from Eq. (5), regresses the action bounding box, and classifies the action. Overall decoder design follows that of YOWO [30]. Latent tensor $\mathbf{Z}_t$ passed through CFAM module [30] to capture inter-channel dependencies and project channel to final output channel. The final output channel is set to the five anchors multiplied by the sum of the number of classes, the number of bounding box elements ($\{x, y, w, h\}$), and their confidence score ($5 \times (\#Class + 5)$). K-means select prior anchors for each dataset. More detail about the decoder is available in the supplementary material.

4.4. Training

This section describes the training scheme and the loss functions. We first present loss functions used to train our model. We define our loss function by combining localization loss and classification loss following [30]. We use Huber loss for each element of the bounding box for action localization ($\mathcal{L}_\text{x}, \mathcal{L}_\text{y}, \mathcal{L}_\text{w}, \mathcal{L}_\text{h}$) and Mean Square Error (MSE) for confidence score ($\mathcal{L}_\text{conf}$). We add all of them together to define localization loss.

$$\mathcal{L}_\text{loc} = \mathcal{L}_\text{x} + \mathcal{L}_\text{y} + \mathcal{L}_\text{w} + \mathcal{L}_\text{h} + \mathcal{L}_\text{conf} \tag{14}$$

We tested feature loss which minimizes MSE between predicted latent feature and encoded feature (MSE$(\mathbf{Z}_t, \hat{\mathbf{Z}}_t)$) following [15]. However, performance is slightly decreased when trained with feature loss. For action classification, we use Focal loss [37].

$$\mathcal{L}_\text{cls} = -\sum_{i=1}^{C}(y_i(1-\hat{y}_i)^\gamma \log(\hat{y}_i) + (1-y)\hat{y}_i{}^\gamma \log(1-\hat{y}_i)), \tag{15}$$

where C is the number of classes, γ is a modulating factor of focal loss, and y_i and $\hat{y}_i$ indicate one hot encoded vector of ground-truth action label and predicted class probability, respectively. Finally, the total loss for training is defined as,

$$\mathcal{L}_\text{total} = \mathcal{L}_\text{loc} + \lambda\mathcal{L}_\text{cls}, \tag{16}$$

where λ is the weighting parameter for classification loss.

For training, we used scheduled sampling [5] following the extrapolation task of Latent ODE [49], that feeds in either the previously observed value ($\mathbf{Z}_t$) or predicted value ($\hat{\mathbf{Z}}_t$) with probability 0.5. When inference, we use values from observation ($\mathbf{Z}_t, t \leq O$) and predicted values for future frames ($\hat{\mathbf{Z}}_t, O \leq t \leq T$) to localize the actions.

5. Experiments

We tested our method on the SALF task and compared the results with other state-of-the-art long-range temporal modeling methods [6, 15, 18, 51]. For evaluation, we adapt frame-level mean Average Precision (Frame-mAP) with IoU threshold 0.5 following previous works [30, 42]. The mAP is calculated as the AUC of the precision-recall curve of detection results from each frame and means over different action categories. We evaluate Frame-mAP on 1) the entire sequence, which includes observed and unseen frames, and 2) unseen frames only.

5.1. Datsaets

UCF101-24 UCF101-24 is a subset of UCF101 [54]. This dataset contains 24 action categories and 3,207 action videos with spatio-temporal bounding box annotations for action tubelets. Most videos have one actor, but

Datasets	Methods	Observation Ratio									
		10%		20%		30%		40%		50%	
		TOTAL	UNSEEN	TOTAL	UNSEEN	TOTAL	UNSEEN	TOTAL	UNSEEN	TOTAL	UNSEEN
UCF101-24	RNN [51]	33.73	29.74	44.17	37.30	49.18	39.29	54.06	42.32	53.92	41.38
	ODE-RNN [6]	-	31.56	-	34.84	-	35.59	-	37.71	-	39.70
	ODE2VAE [67]	-	31.11	-	34.36	-	35.42	-	37.10	-	37.52
	PhyDNet [18]	33.98	29.90	44.63	37.22	49.39	39.69	53.14	41.44	55.56	42.47
	Transformer [15]	37.10	34.21	44.42	37.85	48.86	41.06	52.66	43.73	56.26	44.87
	AdamsFormer (Ours)	**43.28**	**37.86**	**50.04**	**41.00**	**52.82**	**42.92**	**57.03**	**45.25**	**62.21**	**48.74**
JHMDB-21	RNN [51]	13.43	10.85	27.81	24.76	34.71	32.06	32.90	29.82	33.95	31.19
	ODE-RNN [6]	-	19.99	-	21.63	-	24.57	-	28.86	-	31.69
	ODE2VAE [67]	-	13.14	-	23.09	-	26.59	-	33.18	-	29.35
	PhyDNet [18]	1.58	0.80	24.08	22.22	30.96	29.74	30.13	28.85	29.35	29.41
	Transformer [15]	35.20	35.17	39.58	40.24	42.71	44.09	48.87	50.66	47.46	50.45
	AdamsFormer (Ours)	**49.93**	**49.39**	**49.72**	**49.55**	**50.94**	**51.72**	**52.35**	**53.28**	**51.18**	**52.81**

Table 2. Experimental results on UCF101-24 and JHMDB-21 datasets. We evaluate the models in terms of frame-mAP. Bold figures indicate the best performance for each setup. 'TOTAL' and 'UNSEEN' in the table represent localization performance on the total sequence and unseen frames only, respectively.

some have multiple action instances with different spatio-temporal boundaries. For experiments, we stack 8 frames to make a clip and sample every 4 clips from the video to construct an action sequence. We set the total length of the sequence as 20 and used the split 1 for training and testing following previous works. Further implementation details are provided in the supplementary material.

JHMDB-21 This dataset is a subset of HMDB-51 dataset [32], containing 21 action categories in 928 untrimmed action videos. For our experiments, we use the first split of the dataset. We stack 8 frames for the video clip and use all possible subsequent 10 clips in the videos for both training and testing. We discard items that are less than 10 clips.

5.2. Comparison with Other Methods

To investigate the advantage of our proposed method in the SALF task, we compare our approach with existing long-range temporal dependency modeling methods. To validate the effectiveness of temporal modeling in AdamsFormer, we replace the future feature predictor with other temporal modeling methods. For pair comparison, all experimental setups are the same as the setup described in implementation details 5.2.

Baseline Implementations We select five widely-used long-range temporal modeling methods (RNN [51], ODE-RNN [6], ODE2VAE [67], PhyDNet [18], and Transformer [15]) as our baselines: 1) **RNN**: Implemented using a 3-layer Conv-LSTM [51] with a 3×3 kernel, RNN takes $\mathbf{Z}_t$ as input for the observation frames where $t \leq O$ and predicts the next frame $\mathbf{Z}_{t+1}$. The network recursively predicts future latent features by reinjecting the predicted representation as input for unseen frames where $O \leq t \leq T$. 2) **ODE-RNN**: A 3-layer Conv-LSTM with a 3×3 kernel is used as the encoder, taking $\mathbf{Z}_{1:O}$ as input. ODE-RNN [6] then extrapolates the RNN output as an initial value using the ODE solver[1] with the Runge-Kutta 45 method. Since the RNN output represents the entire observed sequence, we do not report action localization scores on observed frames for this method. 3) **ODE2VAE**: In this baseline, we replace the ODE part of our ODE-RNN baseline with ODE2VAE [67]. 4) **PhyDNet**: We use the official implementation[2] provided by the author [18], stacking three layers of each PhyCell and ConvCell. 5) **Transformer**: Following [15], we use GPT-2 [46] to construct the transformer. Additionally, we add a feature loss to Eq. (16), defined as the Mean Square Error between the predicted feature $\hat{\mathbf{Z}}_t$ and the encoded one $\mathbf{Z}_t$.

Results We present overall results in Table 2 that reveal the effectiveness of AdamsFormer. For the 10% observation ratio, we report the performance of AdamsFormer with $N = 2$ for UCF101-24 and $N = 1$ for JHMDB-21, whereas all other setups are following Sec. 5.2. When the number of observation frames is less than the multi-step order, the model can't take enough initial values to solve ODE. In the case of UCF01-24, 10% of 20 frames are two frames smaller than the multi-step order we set for this dataset $N = 4$. In both UCF101-24 and JHMDB-21, our model outperforms all other long-range temporal modeling methods on both observed and unseen frames in every observation ratio with a sizable margin. Significantly, the lower the observation ratio is, the more significant the performance gap is, demonstrating the advantage of our proposed work on long-range temporal modeling. For example, AdamsFormer shows higher performance than other methods in 10% of observation ratio by 8 points in unseen frames compared to PhyDNet. When the observation ratio is low, predicting the location of the future action is extremely difficult since the model needs to predict unseen long-range frames from the few observation. Transformer [15] shows a good performance on UCF101-24 but has poor performance

[1] https://github.com/rtqichen/torchdiffeq.git
[2] https://github.com/vincent-leguen/PhyDNet.git

Methods	Observation Ratio									
	10%		20%		30%		40%		50%	
	TOTAL	UNSEEN	TOTAL	UNSEEN	TOTAL	UNSEEN	TOTAL	UNSEEN	TOTAL	UNSEEN
Single-step ($N = 1$)	40.74	36.81	47.00	39.54	51.42	42.75	55.41	44.41	60.13	47.39
Multi-step ($N = 2$)	**43.28**	**37.86**	46.96	40.04	51.50	41.57	55.15	44.28	59.83	47.10
Multi-step ($N = 3$)	-	-	49.16	**41.15**	52.33	42.54	56.89	45.18	61.32	47.19
Multi-step ($N = 4$)	-	-	**50.04**	41.00	52.82	**42.92**	57.03	45.25	62.21	**48.74**
Multi-step ($N = 5$)	-	-	-	-	53.20	42.23	**58.69**	**45.46**	61.12	46.80
Multi-step ($N = 6$)	-	-	-	-	**53.21**	42.34	57.26	44.66	**63.01**	48.00

Table 3. Comparison of our model performances on different multi-step order setups. When the multi-step order is $N = 1$, it is equivalent to the Euler method, which is single-step.

in low observation ratio on the JHMDB-21 dataset. In contrast, AdamsFormer performs almost similarly in mAP in total sequence with unseen frames in the JHMDB-21 dataset, indicating latent features in observed frames are well-extrapolated to future frames.

5.3. Ablations and Analysis

To examine the effect of individual parameters and components of AdamsFormer, we conduct SALF on different configurations of our model. All experiments in this section are performed on the UCF101-24 dataset.

Advantage of multi-Step method We first validate the effect of our proposed multi-step method for future feature prediction against the single-step method for solving ODE in Table 3. When $N = 1$ in the Adams method, it is equivalent to the Euler method, a single-step approach. We report the results of the Euler method for the single-step method, but most single-step methods (Runge Kutta, MidPoint, and Dormand-Prince) show similar results to Euler. The multi-step method outperforms the single-step method across all observation ratios, confirming its advantage.

Both single-step and multi-step methods are implemented with $N \times 1$ temporal convolution as in Eq. (6). The increase in latency as N grows is negligible due to the following reasons. First, temporal convolution contributes only a tiny fraction of the total computational cost, which is mainly dominated by other elements such as 2D/3D CNNs and Transformers. Second, the latency of temporal convolution usually grows sub-linearly with N due to its efficient GPU implementation.

In Fig. 4, we further compare the action localization results for single-step and multi-step methods. The multi-step method more precisely predicts the future by incorporating information from previous steps rather than discarding it. These results validate our hypothesis that the multi-step method provides robust representation extrapolation against noisy camera motion.

Order of Multi-Step Method To investigate the impact of multi-step order, we compare the frame-mAP of our model using different multi-step orders (N) in Table 3. We evaluate each order in 20% and 50% of observation to see the effect of order in long-term and short-term prediction, respectively. We only report the case when the number of frames is less than the order of the multi-step method. We observe that the model with $N = 4$ shows the best performance for both short-term (50% observation) and long-term (20% observation) action localization. We also see that the performance of unseen frames tends to decrease when N is larger than 4, showing that larger-order does not necessarily improve the performance.

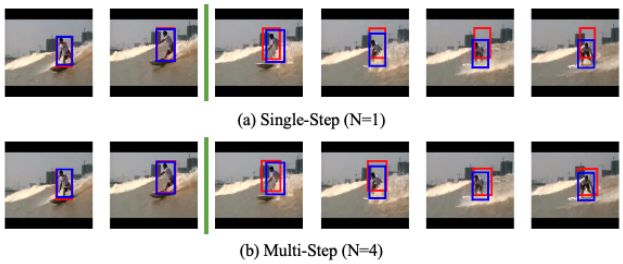


Figure 4. Comparison of action localization results between (a) Single-step and (b) Multi-step methods. The first two are observed frames, and the last four are unseen.

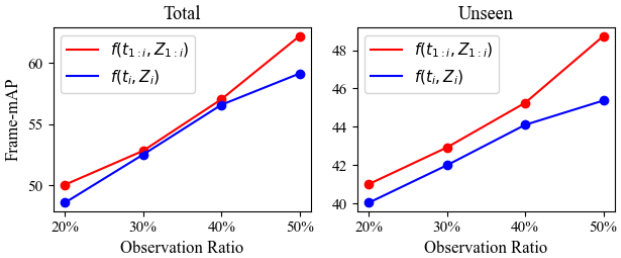


Figure 5. Comparisons between different ODE function designs. The red line shows the results of our model, and the blue line visualizes the results of stacked convolution layers.

ODE Function Configuration We plot the performance of different ODE function ($f(\cdot)$) designs in Fig. 5. We design our ODE function to attend all previous frames $f(t_{1:i}, \mathbf{Z}_{1:i})$ with Transformer with a casual mask as described in Sec. 4.3.2. To justify the design of our ODE function, we set up the baseline by stacking convolution four layers with 3×3 kernel following [43] that only takes current frame information as input ($f(t_i, \mathbf{Z}_i)$). All other setups

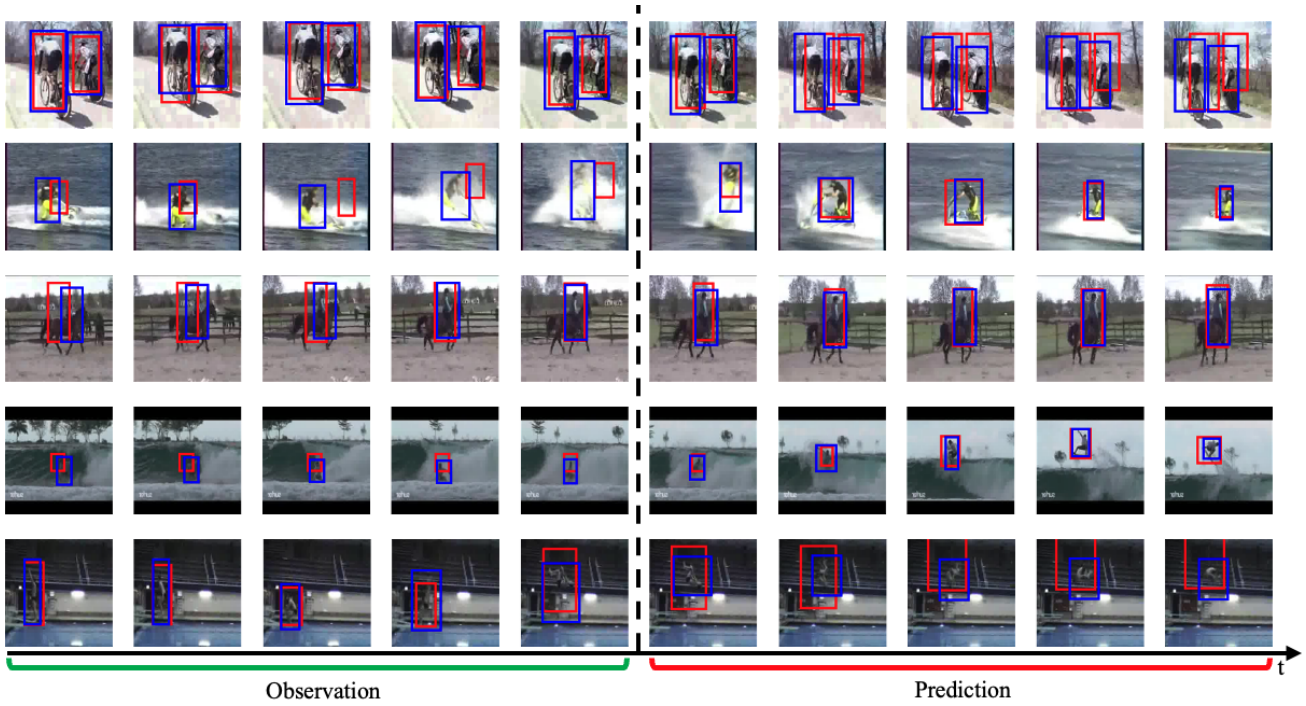


Figure 6. Qualitative results of AdamsFormer on UCF101-24. The red and blue boxes represent prediction and ground truth, respectively. The left five frames are localization results on the observed sequence, whereas the right five frames are those of future frames.

Methods	Observation Ratios			
	20%	30%	40%	50%
w/ obs localization	41.00	42.92	45.25	48.74
w/o obs localization	39.99	41.01	44.10	45.67

Table 4. Comparison of our model performances with and without action localization in the observed frames.

for baseline are the same as described in Sec. 5.2. Our design outperforms a baseline that uses only the current frame for ODE function input. Attending to previous information can provide the ability to distinguish similar scenes utilizing context when modeling the dynamics of latent features.

Action Localization in Observed Frames We compare the performance of our model with and without action localization in the observed frame in Tab. 4. We note that the model trained with action localization in the observed frames performs better than those without it. It is because localizing action with the feature of encoded latent feature from observed frames ($\mathbf{Z}_t, t \leq O$) gives a generalization effect for the decoder regressing the action from the extrapolated latent features ($\hat{\mathbf{Z}}_t, O \leq t \leq T$).

5.4. Qualitative Results

In Fig. 6, we provide qualitative action localization results on the UCF101-24 dataset. These results are the output of AdamsFomer on 50% observation. For simplicity, we visualize the last five frames from the observed and the first five frames of predictions. It shows that AdamsFormer localizes action accurately for both observation and future time horizons. Especially when the action is slow (3rd rows in Fig. 6), our model predicts action location with high accuracy. As shown in the examples of the 2nd, 3rd, and 5th rows in Fig. 6, we also observe that Adamsformer successfully predicts dynamic actions based on the captured movement on observation. Also, since AdamsFormer extrapolates latent features, it can detect multiple actions in the same frame as the example in 1st row of the figure. In a supplementary document, we provide more experimental results of multi-agent and multi-action setups.

6. Conclusion

In this work, we introduce a new task, spatio-temporal action localization in the future (SALF), which aims to localize and classify actions in future frames. To tackle this problem, we propose a novel framework named AdamsFormer that extrapolate observed latent feature to future frames. Through extensive experiments, we prove that AdamsFormer outperforms existing long-range temporal modeling algorithms on SALF.

Acknowledgement We acknowledge US National Science Foundation (FW-HTF 1839971) and the Feddersen Chair Funds for Professor Karthik Ramani. We thank Kumar Akash and the anonymous reviewers for their helpful and constructive comments.

References

[1] Yazan Abu Farha, Alexander Richard, and Juergen Gall. When will you do what?-anticipating temporal occurrences of activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5343–5352, 2018. 2

[2] Nakul Agarwal, Yi-Ting Chen, Behzad Dariush, and Ming-Hsuan Yang. Unsupervised domain adaptation for spatio-temporal action localization. In *BMVC*, 2020. 1, 2

[3] Humam Alwassel, Silvio Giancola, and Bernard Ghanem. Tsp: Temporally-sensitive pretraining of video encoders for localization tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3173–3183, 2021. 1

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5

[5] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015. 5

[6] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018. 2, 3, 5, 6

[7] Hyung-gun Chi, Myoung Hoon Ha, Seunggeun Chi, Sang Wan Lee, Qixing Huang, and Karthik Ramani. Infogcn: Representation learning for human skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20186–20196, 2022. 1

[8] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736, 2018. 2

[9] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *International conference on machine learning*, pages 1174–1183. PMLR, 2018. 2

[10] Haodong Duan, Yue Zhao, Yuanjun Xiong, Wentao Liu, and Dahua Lin. Omni-sourced webly-supervised learning for video recognition. In *European Conference on Computer Vision*, pages 670–688. Springer, 2020. 1

[11] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 1

[12] Yutong Feng, Jianwen Jiang, Ziyuan Huang, Zhiwu Qing, Xiang Wang, Shiwei Zhang, Mingqian Tang, and Yue Gao. Relation modeling in spatio-temporal action localization. *arXiv preprint arXiv:2106.08061*, 2021. 2

[13] Basura Fernando and Samitha Herath. Anticipating human actions by correlating past with the future with jaccard similarity measures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13224–13233, 2021. 1

[14] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Red: Reinforced encoder-decoder networks for action anticipation. *arXiv preprint arXiv:1707.04818*, 2017. 2

[15] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13505–13515, 2021. 1, 2, 5, 6

[16] Dayoung Gong, Joonseok Lee, Manjin Kim, Seong Jong Ha, and Minsu Cho. Future transformer for long-term action anticipation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3052–3061, 2022. 1, 2

[17] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6047–6056, 2018. 2

[18] Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11474–11484, 2020. 2, 5, 6

[19] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. 4

[20] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 5

[21] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[22] Rui Hou, Chen Chen, and Mubarak Shah. Tube convolutional neural network (t-cnn) for action detection in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 5822–5831, 2017. 2

[23] Jian-Fang Hu, Wei-Shi Zheng, Lianyang Ma, Gang Wang, Jianhuang Lai, and Jianguo Zhang. Early action prediction by soft regression. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2568–2583, 2018. 1

[24] De-An Huang and Kris M Kitani. Action-reaction: Forecasting the dynamics of human interaction. In *European Conference on Computer Vision*, pages 489–504. Springer, 2014. 2

[25] Mihir Jain, Jan Van Gemert, Hervé Jégou, Patrick Bouthemy, and Cees GM Snoek. Action localization with tubelets from motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 740–747, 2014. 2

[26] Boyan Jiang, Yinda Zhang, Xingkui Wei, Xiangyang Xue, and Yanwei Fu. Learning compositional representation for 4d captures with neural ode. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5340–5350, 2021. 2

[27] Ming Jin, Yu Zheng, Yuan-Fang Li, Siheng Chen, Bin Yang, and Shirui Pan. Multivariate time series forecasting with dynamic graph neural odes. *IEEE Transactions on Knowledge and Data Engineering*, 2022. 2

[28] Kapil D Katyal, Gregory D Hager, and Chien-Ming Huang. Intent-aware pedestrian prediction for adaptive crowd navigation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3277–3283. IEEE, 2020. 1

[29] Qiuhong Ke, Mario Fritz, and Bernt Schiele. Time-conditioned action anticipation in one shot. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9925–9934, 2019. 2

[30] Okan Köpüklü, Xiangyu Wei, and Gerhard Rigoll. You only watch once: A unified cnn architecture for real-time spatiotemporal action localization. *arXiv preprint arXiv:1911.06644*, 2019. 1, 2, 4, 5

[31] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787, 2014. 2

[32] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE, 2011. 2, 6

[33] Dong Li, Zhaofan Qiu, Qi Dai, Ting Yao, and Tao Mei. Recurrent tubelet proposal and recognition networks for action detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 303–318, 2018. 2

[34] Yixuan Li, Zixu Wang, Limin Wang, and Gangshan Wu. Actions as moving points. In *European Conference on Computer Vision*, pages 68–84. Springer, 2020. 2

[35] Yuxuan Liang, Kun Ouyang, Hanshu Yan, Yiwei Wang, Zekun Tong, and Roger Zimmermann. Modeling trajectories with neural ordinary differential equations. In *IJCAI*, pages 1498–1504, 2021. 2

[36] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3889–3898, 2019. 1

[37] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 5

[38] Bingbin Liu, Ehsan Adeli, Zhangjie Cao, Kuan-Hui Lee, Abhijeet Shenoi, Adrien Gaidon, and Juan Carlos Niebles. Spatiotemporal relationship reasoning for pedestrian intent prediction. *IEEE Robotics and Automation Letters*, 5(2):3485–3492, 2020. 1

[39] Miao Liu, Siyu Tang, Yin Li, and James M Rehg. Forecasting human-object interaction: joint prediction of motor attention and actions in first person video. In *European Conference on Computer Vision*, pages 704–721. Springer, 2020. 2

[40] Shaowei Liu, Subarna Tripathi, Somdeb Majumdar, and Xiaolong Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3282–3292, 2022. 1, 2

[41] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. Ego-topo: Environment affordances from egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 163–172, 2020. 2

[42] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 464–474, 2021. 1, 2, 5

[43] Sunghyun Park, Kangyeol Kim, Junsoo Lee, Jaegul Choo, Joonseok Lee, Sookyung Kim, and Edward Choi. Vid-ode: Continuous-time video generation with neural ordinary differential equation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2412–2422, 2021. 2, 7

[44] Xiaojiang Peng and Cordelia Schmid. Multi-region two-stream r-cnn for action detection. In *European conference on computer vision*, pages 744–759. Springer, 2016. 2

[45] Ruijie Quan, Linchao Zhu, Yu Wu, and Yi Yang. Holistic lstm for pedestrian trajectory prediction. *IEEE transactions on image processing*, 30:3229–3239, 2021. 1, 2

[46] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 6

[47] Amir Rasouli, Iuliia Kotseruba, Toni Kunic, and John K Tsotsos. Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6262–6271, 2019. 1, 2

[48] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 4

[49] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019. 2, 3, 5

[50] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long-range video understanding. In *European Conference on Computer Vision*, pages 154–171. Springer, 2020. 2

[51] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015. 5, 6

[52] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3637–3646, 2017. 1, 2

[53] Lin Song, Shiwei Zhang, Gang Yu, and Hongbin Sun. Tacnet: Transition-aware context network for spatio-temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11987–11995, 2019. 2

[54] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2, 4, 5

[55] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 318–334, 2018. 2

[56] Jiajun Tang, Jin Xia, Xinzhi Mu, Bo Pang, and Cewu Lu. Asynchronous interaction aggregation for action detection. In *European Conference on Computer Vision*, pages 71–87. Springer, 2020. 2

[57] Marike K van den Broek and Thomas B Moeslund. Ergonomic adaptation of robotic movements in human-robot collaboration. In *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pages 499–501, 2020. 1

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 4

[59] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 98–106, 2016. 2

[60] Lihui Wang, Sichao Liu, Hongyi Liu, and Xi Vincent Wang. Overview of human-robot collaboration in manufacturing. In *Proceedings of 5th international conference on the industry 4.0 model for advanced manufacturing*, pages 15–58. Springer, 2020. 1

[61] Tianyi Wang, Xun Qian, Fengming He, Xiyun Hu, Yuanzhi Cao, and Karthik Ramani. Gesturar: An authoring system for creating freehand interactive augmented reality applications. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 552–567, 2021. 1

[62] Xionghui Wang, Jian-Fang Hu, Jian-Huang Lai, Jianguo Zhang, and Wei-Shi Zheng. Progressive teacher-student learning for early action prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3556–3565, 2019. 1

[63] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *Proceedings of the IEEE international conference on computer vision*, pages 3164–3172, 2015. 2

[64] Feng Wen, Zhongda Sun, Tianyiyi He, Qiongfeng Shi, Minglu Zhu, Zixuan Zhang, Lianhui Li, Ting Zhang, and Chengkuo Lee. Machine learning glove using self-powered conductive superhydrophobic triboelectric textile for gesture recognition in vr/ar applications. *Advanced science*, 7(14):2000261, 2020. 1

[65] Song Wen, Hao Wang, and Dimitris Metaxas. Social ode: Multi-agent trajectory forecasting with neural ordinary differential equations. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 217–233. Springer, 2022. 2

[66] Xitong Yang, Xiaodong Yang, Ming-Yu Liu, Fanyi Xiao, Larry S Davis, and Jan Kautz. Step: Spatio-temporal progressive learning for video action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 264–272, 2019. 2

[67] Cagatay Yildiz, Markus Heinonen, and Harri Lahdesmaki. Ode2vae: Deep generative second order odes with bayesian neural networks. *Advances in Neural Information Processing Systems*, 32, 2019. 2, 6

[68] Jiaojiao Zhao and Cees GM Snoek. Dance with flow: Two-in-one stream action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9935–9944, 2019. 2

[69] Jiaojiao Zhao, Yanyi Zhang, Xinyu Li, Hao Chen, Bing Shuai, Mingze Xu, Chunhui Liu, Kaustav Kundu, Yuanjun Xiong, Davide Modolo, et al. Tuber: Tubelet transformer for video action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13598–13607, 2022. 2