Entry-Flipped Transformer for Inference and Prediction of Participant Behavior

Bo Hu[1,2] and Tat-Jen Cham[1,2]

[1] Singtel Cognitive and Artificial Intelligence Lab (SCALE@NTU), Singapore
[2] School of Computer Science and Engineering, Nanyang Technological University, Singapore
{hubo,astjcham}@ntu.edu.sg

Abstract. Some group activities, such as team sports and choreographed dances, involve closely coupled interaction between participants. Here we investigate the tasks of inferring and predicting participant behavior, in terms of motion paths and actions, under such conditions. We narrow the problem to that of estimating how a set target participants react to the behavior of other observed participants. Our key idea is to model the spatio-temporal relations among participants in a manner that is robust to error accumulation during frame-wise inference and prediction. We propose a novel Entry-Flipped Transformer (EF-Transformer), which models the relations of participants by attention mechanisms on both spatial and temporal domains. Unlike typical transformers, we tackle the problem of error accumulation by flipping the order of query, key, and value entries, to increase the importance and fidelity of observed features in the current frame. Comparative experiments show that our EF-Transformer achieves the best performance on a newly-collected tennis doubles dataset, a Ceilidh dance dataset, and two pedestrian datasets. Furthermore, it is also demonstrated that our EF-Transformer is better at limiting accumulated errors and recovering from wrong estimations.

Keywords: Entry-Flipping, Transformer, Behavior Prediction

1 Introduction

The development of computer vision with machine learning has led to extensive progress in understanding human behavior, such as human action recognition and temporal action detection. Although state-of-the-art algorithms have shown promise, a majority of methods have been focused only on individuals without explicitly handling interaction between people. However, human behavior can span a wide range of interaction coupling, from the independence of strangers passing each other, to highly coordinated activities such as in group sports and choreographed dances.

The behavior of a person can be treated as a combination of self intention and social interaction, where the latter is more crucial in group activities. Current group-related computer vision works do not focus much on scenarios with heavy

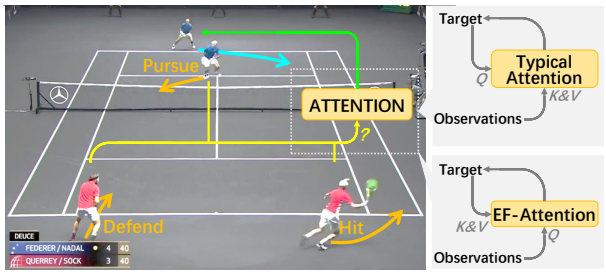


Fig. 1. This paper focuses on participants behavior prediction and inference, where the behavior of target participant from a group activity is estimated with observation of other participants. Entry-Flipping (EF) mechanism is proposed for attention function to obtain accurate prediction and inference by flipping the query, key, and value entries.

social interaction among participants. For example, in pedestrian trajectory prediction [2, 41], the behavior of a pedestrian is based more on self intention than social interaction, with the latter cursorily for avoiding collisions.

To further explore the model of social interactions in group activities, we consider the tasks of inferring and predicting the behavior of some participants as they react to other participants. In these tasks, we hypothesize that the behavior of participants of a group activity are less dependent on self intentions, and instead dominated by how other participants behave. To formalize the problem, we consider a group as split into two sets of observed and target participants. For target participants, we assume that no data is provided beyond some initial states — the objective is thus to infer their behavior based *only* on the continuing data received from observed participants (see fig. 1). We believe that this modeling of reactive human behavior in closely coupled activities such as team sports, will eventually lead to enabling more realistic agent behavior models, *e.g.* for simulation in games or sports training.

The task of inferring or predicting participant behavior is a frame-wise sequence estimation problem. There are many existing models focused on sequence estimation, such as Recurrent Neural Networks (RNN) based methods [26, 33, 22] and attention-based methods [27, 36]. However, these methods face the problem of error accumulation, as the recurrence involves using the output estimation from the previous step as the input in the next step. While this leads to temporally smooth predictions, small errors at each step accumulate over time, leading to large final errors. Taking a typical transformer [27] as an example, the cross-attention in the decoder auto-regressively uses the previous estimate as query input. As the query is the base of an attention function, errors in subsequent queries will often grow, even if the key and value entries are accurate. This may not be a concern for *e.g.* open-ended text generation, but becomes an issue for our tasks that prioritize accurate current estimates over temporal consistency.

In this paper, we propose the Entry-Flipped Transformer (EF-Transformer), a novel framework for the inference and prediction of participant behavior. Two key properties needed are: i) good relation modeling, ii) limiting the error accu-

mulation. To model spatio-temporal relations among all participants in different frames, we adopt a transformer-based structure with multiple layers of encoders and decoders. In every encoder, separate attentions are used for the spatial domain, *i.e.* involving different participants, and the temporal domain, *i.e.* across different frames. Each decoder contains spatio-temporal self-attention and also cross-attention to relate features of observed and target participants. To limit accumulated errors during frame-wise inference and prediction, an entry-flipped design is introduced to the cross-attention in decoders, to focus more on correctness of output than smoothness. In our method, the query, key, and value entries of decoders are flipped *w.r.t.* the typical order. As accurate information of observed participants is sent to query entry of the attention function at each step, error accumulation can be effectively suppressed.

The main contributions of this paper are as follows:

- We articulate the key considerations needed for inferring and predicting participant behavior in group activities that involve highly coupled interactions.
- A novel EF-Transformer framework is proposed for this task, where query, key, value entries are flipped in cross-attention of decoders.
- Our method achieved SOTA performance on a tennis doubles dataset and a Ceilidh dance dataset that involve highly coupled interactions, and also outperformed other methods on looser coupled pedestrian datasets.
- We show our method is more robust at limiting accumulated errors and recovering from spike errors.

2 Related Work

Relation Modeling. Participant behavior prediction involve several modules, with a core of spatio-temporal relation modeling. Probabilistic graphical models have been used to model relations, *e.g.* Dynamic Bayesian Networks (DBN) [40], Conditional Random Fields (CRF) [4], but these models heavily relied on feature engineering. With deep learning, models can directly learn the relations and find good features simultaneously. Convolutional Neural Networks (CNN) are widely employed to extract features from images and videos, while deeper layers of a CNN can be viewed as relation modeling since they summarize features from a larger image region [19, 24, 9, 5]. Graph Convolution Networks (GCN) [35, 39] are used to learn the relation among features without a fixed grid format. However, convolutions usually have limited receptive fields, and are enlarged only through many layers. RNNs, such as LSTM, have been used to model temporal relation in sequences [32, 3]. Different from CNNs processing all entries in one go, RNNs are applied iteratively over time. Attention mechanisms were popularized by the Transformer [27] and became adopted for both spatial and temporal relation modeling [43, 8, 30]. Attention facilitates summarization for different types of input, leading to better generalization, which can be built upon backbone networks [14, 16, 21], or in feature learning [11]. However, the computational cost of attention is large, thus many methods [42, 28, 37] are hybrids involving a combination of CNN, RNN, and attention to balance efficiency and effectiveness.

Group Relevant Tasks. Group activities typically involve significant behavorial relations among group members. Group activity recognition aims to estimate video-level activity labels. In [6, 18, 31, 17] RNN was used to model temporal relation of each person and pooled all persons together for recognition. Cross inference block has been proposed in HiGCIN [34] to capture co-occurrence spatiotemporal dependencies. In the actor transformer [13], the transformer encodes all actors after actor-level features are extracted. These frameworks are impressive but unsuitable for our proposed tasks, as they are not designed for frame-level estimation. Another related task is pedestrian trajectory prediction [20, 23, 29, 33]. The goal is to predict moving trajectories of all pedestrians in future frames with observation of a few past frames, where interaction among pedestrians is the important cue. RNN [2, 7], graph-based technique [36], and attention mechanism [12, 25] have been employed for this task. In [41], LSTMs were used for single pedestrian modeling and an attention-based state refinement module designed to capture the spatial relations among different pedestrians. Graph-based attention has been proposed for spatial relation modeling [36], where the graph is built based on spatial distance among pedestrians. The difference between this task and ours is that the former aims to predict the future based on past observation for all pedestrians, while we focus more on models that can continually predict about how target participants will react to behavior of other observed participants. This is particularly important in activities that have very strongly coupled interactions. Nonetheless, existing methods can be applied to our task with minor modification, as described later.

3 Method

3.1 Problem Definition

Participants behavior inference and prediction are to estimate the behavior of a number of target participants in a group, based on information of other observed participants in that group. Supposed there are N participants in the group and they are divided into two sets, with N_{obs} observed participants and N_{tgt} target participants, where $N{=}N_{\text{obs}}{+}N_{\text{tgt}}$. Given a trimmed video clip with T frames, let $\boldsymbol{x}{=}\{x_{i,t}\}_{i=1:N_{\text{obs}},t=1:T}$ denote the behavior of observed participants, where the behavior comprise positions and action labels. Correspondingly, $\boldsymbol{y}{=}\{y_{i,t}\}_{i=1:N_{\text{tgt}},t=1:T}$ denote the behavior of target participants.

The task is to infer and predict $\{y_{i,t}\}_{i=1:N_{\text{tgt}}}$, starting from known initial states of the target participants, $\{y_{i,1}\}_{i=1:N_{\text{tgt}}}$. The estimation proceeds sequentially in time, where the observable input at time t consists of $\{x_{i,\tau}\}_{i=1:N_{\text{obs}},\tau=1:t+K}$, where K is the number of frames *into the future* beyond t. Here, K can be interpreted as the level of (perfect) human foresight of the target participants in predicting how other participants may behave. As an ML problem, $K{=}0$ corresponds to participants behavior prediction, while it becomes inference for $K{\geq}1$. The inference can be performed in an online manner if $K{=}1$, otherwise it has to be offline or with a delay.

3.2 Typical Transformer

A typical Transformer consists of multiple layers of encoder and decoder. Both encoder and decoder involve three modules: attention function, feed forward network (FFN), and normalization, where attention function is

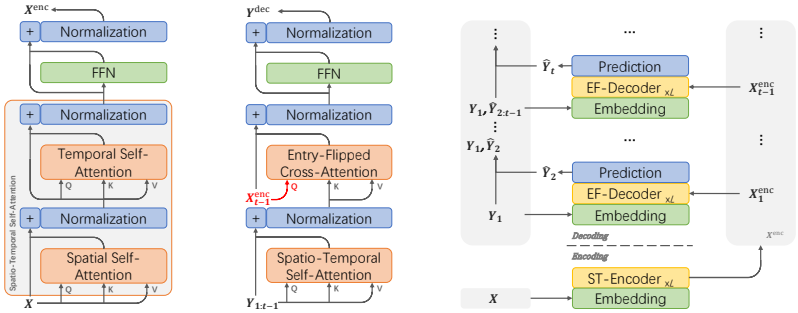
$$X^{\text{att}} = f_o \left[\frac{\boldsymbol{S}\left(f_q\left(X_q\right) f_k\left(X_k\right)^T \right)}{\sqrt{d}} f_v\left(X_v\right) \right] + X_q. \tag{1}$$

In (1), X_q, X_k, and X_v denote the input feature map of query, key, and value correspondingly, and X^{att} is the output attended feature map. $f\left(\cdot\right)$ is the fully-connected (FC) layer, $\boldsymbol{S}\left(\cdot\right)$ is the softmax function on each row of the input matrix, and d is the dimension of X_q and X_k. Noted that multi-head attention scheme in [27] is also employed in all attention modules of our framework, which is ignored in (1) for simplification.

A typical transformer [27] can fit the proposed task, since the feature of observed and target participants can be treated as two different sequences. Compared with machine translation, the observed participants sequence plays the role of source language sentence and the target participants sequence plays the role of target language sentence. However, a typical transformer has a drawback that leads to error accumulation in the task of participant behavior inference and prediction. The attention function (1) takes some other feature (key and value) into consideration when maps the input (query) to the output. From another view, the attention function can be seen as a summarization of the three entries. Different from convolutions or MLP, the three entries play different roles in the attention function. Specifically, the query is the base in the attention function while key and value are the references. In the inference stage, the query of decoder comes from the previous frame estimation, which is not accurate. With a noisy or wrong query entry, it is difficult to recover the feature and provide a relative correct estimation in the next frame. Therefore, the error will accumulate over time, which may not be as relevant in open-ended tasks, *e.g.* text generation.

3.3 Entry-Flipped Transformer

To solve the error accumulation problem, an EF-Transformer is proposed. In our EF-Transformer, encoders apply spatio-temporal attention modules to encode the information from multiple participants in the whole clip. Different from typical transformers, the decoder in EF-Transformer takes the output of the encoder as the query entry. Since this does not depend as much on predictive accuracies in previous frames, it reduces the accumulation of errors. With the Spatio-Temporal Encoder (ST-Encoder) and Entry-Flipped Decoder (EF-Decoder), the proposed EF-Transformer is designed to predict the behavior of target participants frame-by-frame more from observations rather than earlier predictions.



(a) ST-Encoder (b) EF-Decoder (c) Prediction

Fig. 2. The framework of encoder, decoder, and prediction process in the proposed EF-Transformer. For participants inference, X_t^{enc} is sent to decoder to estimate $\hat{Y}_t$.

Spatio-Temporal Encoder An ST-Encoder employs two self-attention functions and an FFN to map the features of observed participants $\boldsymbol{x}$ to encoded features $\boldsymbol{x}^{\text{enc}}$, as shown in Fig. 2(a). Different from word sequences, there are both spatial and temporal domains in each video clip. As the attention function has a quadratic time complexity of input size [27], the time complexity of an attention function over the combined spatio-temporal domain is $\mathcal{O}\left(N^2T^2\right)$. To reduce this, the attention over the two domains are handled separately. Spatial self-attention captures the relation among all participants in one frame, where every frame is sent to spatial self-attention independently. Subsequently, temporal self-attention captures the relation among all time frames for each participant to get the attended feature $\boldsymbol{x}^{\text{att}}$, so that different participants across different time frames are not directly attended. By dividing the self-attention of observed participants into two domains, the time complexity is reduced to $\mathcal{O}(NT(N+T))$. Masked attention [27] is applied to avoid attending the feature beyond K frames. Following [27], a simple FFN is connected to the output of self-attention to obtain $\boldsymbol{x}^{\text{enc}}$ from $\boldsymbol{x}^{\text{att}}$.

Entry-Flipped Decoder In the decoding stage, an EF-Decoder module is introduced. This consists of a self-attention function, a cross-attention function, and an FFN. The self-attention in EF-Decoder is also divided into spatial and temporal domains, which has the same structure as ST-Encoder. It provides the self-attended feature of target participants $\boldsymbol{y}^{\text{att}}$. Unlike in a typical transformer, cross-attention in the proposed EF-Decoder uses as query the encoded features of observed participants, while key and value entries are self-attended features of target participants, including both those initially observed and later predicted. This is shown in Fig. 2(b). Specifically, when predicting frame τ, $\{x_{i,\tau-1}^{\text{enc}}\}_{i=1:N_{\text{obs}}}$ is the query entry and $\{y_{i,t}^{\text{att}}\}_{i=1:N_{\text{tgt}},t=1:\tau-1}$ form the key and value entries. The key idea is that the *query only contains observed participants in the current frame*, which becomes the base for next frame inference or prediction. Keys and

values only contain target participants in past frames, forming the reference bases for next frame inference or prediction. The decoded feature $\boldsymbol{y}^{\text{dec}}$ comes from an FFN stack on the cross-attention function, which is the same as the ST-Encoder.

Justification of Entry Flipping. Why is this difference between our method and a typical transformer important? For NLP translation, the most crucial word usually is the last translated word. Hence, a typical transformer uses the last translated word in the target language as the query entry of cross-attention in the decoder. However, *in scenarios where the behavior of participants are highly coupled and reactive*, such as in game sports, the most important clue for determining the behavior of a target participant in next frame would *not be the past frames of the participant*, but rather the status of *other observed participants in the current frame*. For example, the ideal movement of a tennis player highly depends on the evolving positions of her teammate and opponents, whereas rapid acceleration and direction changes mean that the historical positions of this player is not that critical as a predictor. Therefore entry-flipping is more appropriate for the proposed group behavior inference and prediction tasks.

Prediction Framework The whole prediction (Fig. 2(c)) network includes several layers: i) feature embedding layer, ii) ST-Encoder layers, iii) EF-Decoder layers, and iv) prediction layer.

Feature Embedding. Two FC layers are separately applied on the two types of input, *i.e.* 2D coordinates and action labels of participants, to map to higher dimensional features. We first expand the 2D coordinates $(u_{i,t}, v_{i,t})$, to a normalized 7D geometric feature $x^{\text{g}}_{i,t}$ by

$$x^{\text{g}}_{i,t} = \left[uv_{i,t}, uv^{\Delta}_{i,t}, uv^{R}_{i,t}, t/T\right]^{T}, \tag{2}$$

where

$$\begin{aligned} uv_{i,t} &= \left[\frac{u_{i,t}}{w}, \frac{v_{i,t}}{h}\right], \\ uv^{\Delta}_{i,t} &= \left[\frac{u_{i,t} - u_{i,t-1}}{w}, \frac{v_{i,t} - v_{i,t-1}}{h}\right], \\ uv^{R}_{i,t} &= \left[\frac{u_{i,t} - u_{i,1}}{w}, \frac{v_{i,t} - v_{i,1}}{h}\right] \end{aligned} \tag{3}$$

for a video frame of width w and height h, for which $x^{\text{g}}_{i,t}$ contain absolute coordinates, relative coordinates, and temporal positions, all of which are normalized. $x^{\text{g}}_{i,t}$ is sent to a FC layer f_g to obtain higher dimensional geometric features. Action labels are first converted to one-hot $x^{\text{s}}_{i,t}$, followed by another FC layer f_s. Both types of features are concatenated before positional encoding $x^{\text{pe}}_{i,t}$ [27] is added. Thus, the feature of a participant is

$$x_{i,t} = \left[f_g(x^{\text{g}}_{i,t}), f_s(x^{\text{s}}_{i,t})\right]^{T} + x^{\text{pe}}_{i,t}. \tag{4}$$

Encoders and Decoders. L layers of ST-Encoder and EF-Decoder are stacked. The encoded feature of observed participants from output of last layer ST-Encoder is used as the query entry of all layers of EF-Decoder. The last EF-Decoder layer output is the feature that ready for target participants inference and prediction.

Prediction. A prediction layer provides a mapping of $\mathbb{R}^{N_{\text{obs}} \times D} \mapsto \mathbb{R}^{N_{\text{tgt}} \times D_{\text{out}}}$, where D is the feature dimension of one participant in one frame. The features of N_{obs} observed participants are flattened before inference or prediction. D_{out} is dimension of output, which is 2 for trajectory estimation and number of action categories for action classification. The prediction layer consists of three FC layers, where every layer is followed by a nonlinear layer (LeakyReLU in our experiment) except the last. More implementation details can be found in the supplementary.

Loss function. This is a simple L2 loss applied to both trajectory and action estimation:

$$L = \sum_{i=1}^{N_{\text{tgt}}} \sum_{t=2}^{T} \left\| x_{i,t}^{\text{g}^*} - \hat{x}_{i,t}^{\text{g}^*} \right\|_2 + \lambda \left\| x_{i,t}^{\text{s}} - \hat{x}_{i,t}^{\text{s}} \right\|_2 , \tag{5}$$

where $x_{i,t}^{\text{g}^*}$ excludes the temporal coordinates t/T in $x_{i,t}^{\text{g}}$ of (2). In all our experiments, λ=0.1.

4 Experiments

4.1 Datasets and Metrics

We selected three datasets with closely coupled behavior in experiments.

Tennis Dataset A new tennis doubles dataset was collected to evaluate our method. There are 12 videos of whole double games with resolution of 1280×720. 4905 10-frame clips were collected in total, which are downsampled to 2.5 fps and stabilized to remove camera motion. Individual-level bounding boxes and action labels were annotated, with the bottom-center point of each box representing the spatial location of the player. Coarse spatial positions of the ball were also estimated. As it is difficult to determine due to extreme motion blur when the ball was traveling fast, the ball position was only coarsely estimated by spatio-temporal linear interpolation between the locations of two players consecutively hitting the ball. Detailed information of the tennis dataset can be found in the supplemental material. In our experiments, the top-left player was selected as the target participant during testing, while the other three players and the ball were treated as observed entities.

Dance Dataset The dance dataset [1] contains 16 videos from overhead view of Ceilidh dances by two choreographers, where every dance was performed by 10 dancers. Two videos for each choreographer were selected for testing and others for training. The raw video is 5 fps and resolution is 640×480. Here 3754 10-frame clips were collected. The action labels are defined as 'stand', 'walk left', 'walk right', 'walk up','walk down', and 'twirling'. No explicit information about the choreographer was provided during training.

NBA Dataset NBA dataset [38] contains players and ball tracking data from basketball games. During pre-processing, frame rate was down-sampled to 6.25 fps and a subset of over 4000 10-frame clips was built. As actions are not provided in this dataset, we simply assigned 'defensive' and 'offensive' to players as action labels. During training, one defensive player is randomly selected as the target participant, while the first defensive player in the list is selected in testing. The 'resolution' (or court size) in this dataset is 100×50.
Pedestrian Datasets ETH [20] and UCY [23] datasets are conventionally used in pedestrian trajectory prediction. Target participants were randomly selected in training, and the one with longest trajectory was picked in testing. Four nearest neighbors of the target pedestrian among all frames were selected as observed participants. We follow the leave-one-out evaluation in [15].
Metrics To evaluate the accuracy of trajectory inference and prediction, two metrics were computed following [41]: Mean Average Displacement (MAD) is the mean distance between estimation and ground truth over all frames. Final Average Displacement (FAD) is the distance between estimation and ground truth of the last frame. Besides, metrics of short, middle, and long trajectory lengths were computed separately, where the length threshold was statistically determined over all samples to even out the number of samples across each category. For action inference and prediction, Macro F1-scores are reported.

4.2 Baseline and Other SOTA Methods

We compare with several methods in our experiments:
CNN-based Method. This framework is based on spatial and temporal convolutional layers. The encoder consists of 2 convolutional layers while the decoder consists of 3 convolutional layers. A 5-frame sliding window is applied for input.
RNN-based Method. This framework has encoders and decoders based on two GRU layers. At each frame, the output of the encoder is concatenated with the historical data of target participants before sending to the decoder.
Typical Transformer. The typical transformer [27] here uses the ST-encoder and a typical decoder structure, with an additional future mask added to the attention function of encoding stage.
Pedestrian Trajectory Prediction Methods. [36, 41] are also compared. Modifications are made to apply them to our tasks: i) ground truth of observed pedestrians are provided for all frames in the testing stage, ii) if $K>0$, a K-frame time shift over target participants is adopted to ensure the network has correct information of K-frame future of observed participants.

4.3 Ablation Study

In this section, we compare several ST-Encoder structures. $\mathbf{S}+\mathbf{T}$ represents the parallel structure, where spatial and temporal self-attentions are operated on separately, with the outputs added together. $\mathbf{S}\rightarrow\mathbf{T}$ and $\mathbf{T}\rightarrow\mathbf{S}$ are sequential structures with different order of spatial and temporal domain. $\mathbf{S}\times\mathbf{T}$ represents

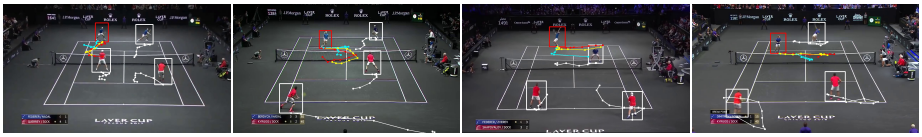


Fig. 3. Visualization of trajectory prediction results of EF-Transformer and typical transformer on tennis dataset. White rectangles and trajectories are the observed participants. Red rectangles are target participants with red trajectories for ground truth. Cyan trajectories are predicted by typical transformer and yellow ones are predicted by our method. Please zoom in to see details.

Table 1. Comparisons of different ST-Encoders and prediction types on Tennis dataset.

Encoder	Pred	MAD				FAD			
		Short	Mid	Long	Avg	Short	Mid	Long	Avg
S+T		**18.70**	31.27	44.51	28.93	**32.89**	51.56	69.15	47.74
T→S	uv^R	19.49	31.01	45.71	29.29	35.31	50.96	69.81	48.43
S×T		19.72	32.05	43.73	29.53	36.28	54.09	67.90	49.95
S→T	uv	40.52	50.42	62.73	48.89	36.11	49.05	64.33	46.91
	Σuv^Δ	20.72	32.91	49.05	31.18	40.12	57.81	78.98	54.93
	uv^R	19.40	**30.04**	**43.04**	**28.35**	35.38	**48.62**	**64.23**	**46.43**

jointly computing attention functions over spatial and temporal domain. In addition, we evaluated the accuracy of different position estimators from among the 3 predicted components in (3), which have overlapping redundancy. Here, the frame-wise relative component uv^Δ is cumulatively summed to get position estimates Σuv^Δ, relative to target positions in the last fully-observed frame. Results are shown in Table 1.

Of the three components, uv^R appeared to be better predicted than the other two. Prediction of absolute coordinates uv is more difficult than only predicting the difference. However, predicting the difference of neighboring frames uv^Δ suffers from error accumulation. The output of frame t have to compensate for the error in predicting frame $t-1$, which can lead to unstable oscillations. Compared with Parallel ST-Encoder, Sequential ST-Encoder achieved better performance except on short trajectories. This is because the query of Sequential ST-Encoder is capable of attending to all other participants in all frames, while query of parallel encoders can only attend the same participants in different frames, or other participants in the same frame. Based on the results above, only predictions of uv^R are reported in the following experiments.

4.4 Trajectory Inference and Prediction

Here we focus solely on trajectory estimation, so ground truth action labels were provided for target participants. Table 2 presents the results of behavior prediction and inference on the tennis and NBA datasets. For the tennis dataset, it

Table 2. Comparisons of trajectory inference and prediction with baselines and SOTA methods on tennis dataset and NBA dataset.

		Methods	MAD				FAD			
			Short	Mid	Long	Avg	Short	Mid	Long	Avg
Inference	Tennis	CNN-based	22.61	41.63	64.43	38.54	42.97	73.27	102.78	67.22
		RNN-based	22.62	41.27	72.88	39.78	38.07	67.86	103.47	63.01
		Transformer	21.17	32.91	46.67	30.95	37.14	52.06	68.14	49.34
		SR-LSTM [41]	21.22	34.46	55.60	33.19	41.49	58.50	90.08	57.60
		STAR [36]	20.28	35.21	55.36	33.16	36.86	57.52	90.01	55.45
		EF-Transformer	**19.40**	**30.04**	**43.04**	**28.35**	**35.38**	**48.62**	**64.23**	**46.43**
Prediction	Tennis	CNN-based	22.58	41.81	71.57	39.80	38.84	70.35	105.26	64.76
		RNN-based	23.84	41.99	78.97	41.57	41.34	68.29	110.63	65.58
		Transformer	20.14	33.09	50.70	31.33	35.85	52.55	71.57	49.67
		SR-LSTM [41]	20.43	43.86	85.88	42.37	39.11	75.36	117.43	69.25
		STAR [36]	23.83	43.80	83.65	43.20	37.83	70.61	117.19	66.50
		EF-Transformer	**19.24**	**30.71**	**41.98**	**28.44**	**34.97**	**50.36**	**62.60**	**46.83**
	NBA	Transformer	1.78	4.25	10.13	3.99	2.91	7.33	18.14	6.93
		SR-LSTM [41]	2.84	4.78	10.53	4.77	6.00	8.90	18.63	9.08
		STAR [36]	4.51	5.96	**10.04**	5.92	5.81	8.81	18.07	8.86
		EF-Transformer	**1.65**	**4.18**	10.05	**3.89**	**2.69**	**7.23**	**18.00**	**6.75**

can be observed that our EF-Transformer achieved the best performance among compared methods, in particular significantly outperforming other methods for long trajectories. Longer trajectories provide greater risk of larger estimation errors, and our entry-flipping mechanism is effective for limiting error accumulation. Performance of SR-LSTM [41] is affected by the limited initial ground truth sequence of target participants to adequately bootstrap the LSTM cell states. Furthermore, estimated coordinates of target participants are sent to the state refinement module, so the hidden state of observed participants may become affected by past estimation errors. Similarly, STAR [36] models the spatial relations of all participants together, where the features of observed participants will also become conflated with inferred features of target participants. Comparing inference and prediction, prediction is harder for all methods as no future information of observed participants is provided. This is especially in the tennis dataset, where the behavior of target participants involve quick reactions to observed participants, often with anticipatory foresight. Some visualizations are shown in Fig. 3, illustrating that our method can predict better trajectories than a typical transformer.

In the NBA dataset, EF-Transformer also outperformed other methods except for the MAD of long trajectories, where STAR [36] surpassed ours only by a tiny 0.01. It can be observed from Table 2 that the performance differences among compared methods are less than for the tennis dataset. We believe the main reason is that in the most of the cases, a defensive player only needs to follow the corresponding offensive player, which is a simpler reaction than the tennis scenario and usually results in a small displacement during prediction for all methods.

Table 3. Comparisons of trajectory prediction with 1 and 2 target participants with baselines and SOTA methods on dance dataset.

	Methods	MAD				FAD			
		Short	**Mid**	**Long**	**Avg**	**Short**	**Mid**	**Long**	**Avg**
$N_{\mathbf{tgt}}$=1	CNN-based	6.91	12.19	14.58	11.13	8.64	14.49	16.86	13.22
	RNN-based	8.60	15.09	20.52	14.61	10.71	17.08	20.69	16.03
	Transformer	7.29	12.75	17.33	12.35	9.63	14.83	19.43	14.53
	SR-LSTM [41]	9.50	15.67	22.48	15.76	11.56	18.16	21.82	17.05
	STAR [36]	9.25	15.34	22.34	15.52	11.76	18.93	23.70	17.99
	EF-Transformer	**6.28**	**9.99**	**12.11**	**9.39**	**7.42**	**10.83**	**12.56**	**10.20**
$N_{\mathbf{tgt}}$=2	CNN-based	7.24	12.55	14.99	11.49	8.78	14.86	16.97	13.42
	RNN-based	9.20	15.77	20.93	15.17	11.56	17.72	21.47	16.79
	Transformer	7.02	12.26	17.49	12.15	9.39	15.50	20.06	14.86
	SR-LSTM [41]	9.19	13.92	18.21	13.68	10.69	15.09	18.11	14.54
	STAR [36]	8.26	14.78	22.77	15.14	10.39	17.07	23.34	16.80
	EF-Transformer	**6.80**	**10.19**	**12.23**	**9.67**	**8.22**	**11.52**	**13.60**	**11.05**

For the dance dataset, we evaluated the methods on a prediction task with different numbers of target participants. Results are listed in Table 3. Our method outperformed all compared methods. It can also be observed that the results of N_{tgt}=2 are comparable to N_{tgt}=1. Although fewer observed participants may make the prediction more difficult, it is possible that having more target participants during training provide better guidance to the network, so that the patterns of the dances are better learned. More results of inference task can be found in the supplemental material.

To evaluate the performance in pedestrian datasets, we follow the setting in [15] to provide 8-frame ground truth for the target participant, as the behavior of a pedestrian highly relies on self intention, which underlies one's historical trajectory. The results are shown in Table 4. Our method achieved the best performance among compared methods. As before, existing methods [41, 36] are not appropriately designed for scenarios with different sets of observed and target participants, conflating accurate observations with inaccurate past estimates. Behavior prediction with 1-frame observation for the target is also evaluated. Results and visualizations can be found in the supplementary.

Table 4. Comparisons of trajectory prediction with baselines and SOTA methods on pedestrian dataset.

Methods	Performance MAD/FAD					
	ETH	HOTEL	ZARA	ZARA2	UNIV	AVG
SR-LSTM [41]	1.09/1.76	0.69/1.31	0.79/1.70	0.88/1.85	1.23/2.32	0.94/1.79
STAR [36]	1.09/2.85	0.69/1.41	0.91/2.08	1.27/2.92	1.00/2.18	0.99/2.23
Transformer	0.73/1.40	0.52/0.93	0.63/1.24	0.68/1.46	1.00/1.96	0.71/1.40
EF-Transformer	**0.70/1.33**	**0.49/0.84**	**0.53/1.07**	**0.54/1.10**	**0.89/1.75**	**0.63/1.22**

Table 5. Comparisons of multi-task prediction with baselines and SOTA methods on dance dataset.'Traj' represents the task of trajectory prediction, during which ground truth action labels are provided. 'Multi' represents the task of multi-task prediction, where both trajectories and action labels have to be predicted.

	Methods	MAD				FAD			
		Short	**Mid**	**Long**	**Avg**	**Short**	**Mid**	**Long**	**Avg**
Traj	Transformer	7.29	12.75	17.33	12.35	9.63	14.83	19.43	14.53
	EF-Transformer	**6.28**	**9.99**	12.11	**9.39**	**7.42**	**10.83**	12.56	**10.20**
Multi	Transformer	7.91	14.73	19.24	13.82	10.77	17.86	21.94	16.72
	EF-Transformer	6.98	10.31	**11.80**	9.63	8.28	11.65	**12.51**	10.75

4.5 Multi-Task Inference and Prediction

In multi-task inference and prediction, trajectories and action labels are estimated simultaneously. Different from previous experiments, estimated action labels are sent to feature embedding for next-frame inference or prediction. We only compare to a typical transformer on dance dataset here. As action labels are very tightly coupled between observed and target players in tennis, it turned out that both methods resulted in 100% action classification and only minor differences to trajectory prediction in Table 2, hence results are placed in the supplemental material.

Trajectory prediction results are shown in Table 5. Without ground truth action labels for target participants, our method achieved comparable trajectory prediction performance to results with ground truth input. In contrast, the typical transformer had worse performance when action labels for target participants had to be estimated. Action prediction confusion matrices are provided in the supplementary. The macro F1-score of our method and typical transformer are 0.99 and 0.90 correspondingly. As our method is capable of limiting accumulated errors, trajectory and action predictions occur in a virtuous cycle, where error robustness in the previous step improves action classification, which in turn improves trajectory prediction. This contrasts with a typical transformer, where error drift leads to poorer action classification and larger errors in trajectory prediction.

4.6 Robustness Analysis

Robustness reflects the ability to limit error accumulation, as well as to recover from large errors (*e.g.* due to sensing failure). To evaluate robustness, the 6D prediction of one middle frame is replaced by a large noise spike of [1,1,-1,-1,-1,-1]. FAD was then computed to compare how well the methods recovered from the spike. This experiment was performed with the inference task on the tennis dataset, where the spike was added to different frames.

Table 6. Comparisons of FAD on tennis dataset with noise involved in different frames.

Noise Position	Transformer FAD				EF-Transformer FAD			
	Short	Mid	Long	Avg	Short	Mid	Long	Avg
No Noise	37.14	52.06	68.14	49.34	35.38	48.62	64.23	46.43
Noise@t=3	75.99	103.24	141.06	99.67	37.23	56.37	84.65	54.15
Noise@t=6	80.03	105.35	145.39	105.85	55.19	64.90	90.71	65.68
Noise@t=9	131.76	161.07	205.26	157.81	115.93	123.30	145.31	124.29

Table 6 shows that both methods can recover from the spike to some extent, noting that better recovery was made by the final frame for earlier spikes. Nonetheless, our method performed significantly better than the typical transformer. Even with a frame 9 spike (second-last frame), our method's FAD increased only about 78 pixels, compared to 108 pixels for the typical transformer.

4.7 Limitations

Our method assumes that a group has a fixed number of participants, all with strongly coupled behavior. Thus in *e.g.* a pedestrian scenario with varying numbers of individuals, not all of whom have correlated behavior, we need to select a fixed number of the most likely related individuals as observations for each target pedestrian (*e.g.* with k-nearest-neighbor filtering). Furthermore, although pedestrian trajectories are smoother than in tennis and dance, it turned out that prediction is also more difficult for our method. This is likely due to less behavioral coupling among pedestrians. When observations are not as informative, our method was predominantly trying to do some form of dead reckoning like other methods, which is difficult to be accurate especially for longer intervals.

5 Conclusion

In this paper, we proposed the EF-Transformer for behavior inference and prediction of target participants based on other observed participants. In our decoder, the order of query, key, and value entries of the cross-attention are flipped to effectively reduce error accumulation. EF-Transformer is evaluated in several experiments, where it outperformed all compared methods on the tennis, dance datasets and pedestrian datasets. Moreover, we demonstrate superior robustness to noise spikes. The framework of EF-Transformer can be used for application to learning realistic agent-based behavior in the future.

Acknowledgements This study is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from Singapore Telecommunications Limited (Singtel), through Singtel Cognitive and Artificial Intelligence Lab for Enterprises (SCALE@NTU).

References

1. Aizeboje, J.: Ceilidh dance recognition from an overhead camera. Msc Thesis of University of Edinburgh (2016)
2. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 961–971 (2016)
3. Aliakbarian, M.S., Saleh, F., Salzmann, M., Fernando, B., Petersson, L., Andersson, L.: Encouraging lstms to anticipate actions very early. In: IEEE International Conference on Computer Vision (ICCV). pp. 280–289 (2017)
4. Amer, M.R., Lei, P., Todorovic, S.: Hirf: Hierarchical random field for collective activity recognition in videos. In: European Conference on Computer Vision (ECCV). pp. 572–585. Springer (2014)
5. Azar, S.M., Atigh, M.G., Nickabadi, A., Alahi, A.: Convolutional relational machine for group activity recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7892–7901 (2019)
6. Bagautdinov, T., Alahi, A., Fleuret, F., Fua, P., Savarese, S.: Social scene understanding: End-to-end multi-person action localization and collective activity recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
7. Becker, S., Hug, R., Hubner, W., Arens, M.: Red: A simple but effective baseline predictor for the trajnet benchmark. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops. pp. 0–0 (2018)
8. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020)
9. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
10. Chen, J., Bao, W., Kong, Y.: Group activity prediction with sequential relational anticipation model. In: European Conference on Computer Vision. pp. 581–597. Springer (2020)
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
12. Fernando, T., Denman, S., Sridharan, S., Fookes, C.: Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection. Neural networks **108**, 466–478 (2018)
13. Gavrilyuk, K., Sanford, R., Javan, M., Snoek, C.G.: Actor-transformers for group activity recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 839–848 (2020)
14. Girdhar, R., Carreira, J., Doersch, C., Zisserman, A.: Video action transformer network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 244–253 (2019)
15. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2255–2264 (2018)
16. Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y.: Relation networks for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)

17. Ibrahim, M.S., Mori, G.: Hierarchical relational networks for group activity recognition and retrieval. In: Proceedings of the European conference on computer vision (ECCV). pp. 721–736 (2018)
18. Ibrahim, M.S., Muralidharan, S., Deng, Z., Vahdat, A., Mori, G.: A hierarchical deep temporal model for group activity recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1971–1980 (2016)
19. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 156–165 (2017)
20. Lerner, A., Chrysanthou, Y., Lischinski, D.: Crowds by example. Computer Graphics Forum **26**(3), 655–664 (2007). https://doi.org/https://doi.org/10.1111/j.1467-8659.2007.01089.x
21. Li, J., Liu, X., Zhang, W., Zhang, M., Song, J., Sebe, N.: Spatio-temporal attention networks for action recognition and detection. IEEE Transactions on Multimedia **22**(11), 2990–3001 (2020)
22. Liang, J., Jiang, L., Carlos Niebles, J., Hauptmann, A.G., Fei-Fei, L.: Peeking into the future: Predicting future person activities and locations in videos. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
23. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: 2009 IEEE 12th International Conference on Computer Vision. pp. 261–268. IEEE (2009)
24. Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3d residual networks. In: IEEE International Conference on Computer Vision (ICCV). pp. 5534–5542. IEEE (2017)
25. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., Savarese, S.: Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1349–1358 (2019)
26. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. arXiv preprint arXiv:1409.3215 (2014)
27. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (NIPS). pp. 6000–6010 (2017)
28. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: ICLR (2018)
29. Vemula, A., Muelling, K., Oh, J.: Social attention: Modeling attention in human crowds. In: 2018 IEEE international Conference on Robotics and Automation (ICRA). pp. 4601–4607. IEEE (2018)
30. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
31. Wu, J., Wang, L., Wang, L., Guo, J., Wu, G.: Learning actor relation graphs for group activity recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9964–9974 (2019)
32. Xu, M., Gao, M., Chen, Y.T., Davis, L.S., Crandall, D.J.: Temporal recurrent networks for online action detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5532–5541 (2019)
33. Xu, Y., Piao, Z., Gao, S.: Encoding crowd interaction with deep neural network for pedestrian trajectory prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5275–5284 (2018)

34. Yan, R., Xie, L., Tang, J., Shu, X., Tian, Q.: Higcin: Hierarchical graph-based cross inference network for group activity recognition. IEEE transactions on pattern analysis and machine intelligence (2020)
35. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proceedings of the AAAI conference on artificial intelligence (2018)
36. Yu, C., Ma, X., Ren, J., Zhao, H., Yi, S.: Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In: ECCV (2020)
37. Yuan, Y., Liang, X., Wang, X., Yeung, D.Y., Gupta, A.: Temporal dynamic graph lstm for action-driven video object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 1801–1810 (2017)
38. Yue, Y., Lucey, P., Carr, P., Bialkowski, A., Matthews, I.: Learning fine-grained spatial models for dynamic sports play prediction. In: 2014 IEEE international conference on data mining. pp. 670–679. IEEE (2014)
39. Zeng, R., Huang, W., Tan, M., Rong, Y., Zhao, P., Huang, J., Gan, C.: Graph convolutional networks for temporal action localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7094–7103 (2019)
40. Zeng, Z., Ji, Q.: Knowledge based activity recognition with dynamic bayesian network. In: European Conference on Computer Vision (ECCV). pp. 532–546. Springer (2010)
41. Zhang, P., Ouyang, W., Zhang, P., Xue, J., Zheng, N.: Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In: CVPR (2019)
42. Zhao, R., Wang, K., Su, H., Ji, Q.: Bayesian graph convolution lstm for skeleton based action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6882–6892 (2019)
43. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: ICLR (2021)