Entry-Flipped Transformer for Inference and Prediction of Participant Behavior (Supplemental Material)

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`

A Overview

We first introduce the implementation details of our framework in Section B and some detailed information of datasets in Section C, especially how the actions are defined. Then the supplementary experiment results are reported in Section D, followed by visualizations on the used datasets (Section E).

B Implementation Details

In our proposed framework, geometric and semantic inputs are mapped to 64D. L=2 layers of ST-Encoder and EF-Decoder are stacked. Inside each encoder and decoder, layer normalization is applied after every attention function and feed forward network (FFN). Multi-head attention is used with H=8 heads. In each attention function, the linear transformation of all query, key, and value entries are set to $\mathbb{R}^{128} \mapsto \mathbb{R}^{8}$. In the experiments on tennis and dance datasets, both trajectories and action labels are the inputs to the network, while for the pedestrian dataset, only trajectories are input. Both prediction (K=0) and inference (K=1) tasks were evaluated. During training, the Adam optimizer is adopted with an initial learning rate of 0.001.

C Details of the Datasets

C.1 Actions in Tennis Dataset

In the self-collected tennis doubles dataset, seven individual-level actions were labeled, which are shown in Table 1. Actions 1 to 3 are performed in the serving stage of a tennis game, while actions 4 to 6 take place after the serving stage. As ball positions are coarsely estimated, the ‘action’ label ‘7’ is assigned to the ball to simplify the implementation.

Table 1. Action labels in tennis dataset

Action Label	Action	Abbr.
1	Serve ball	S
2	Waiting for teammate to serve ball	WTS
3	Waiting for opponent to serve ball	WOS
4	Pursue and hit ball	P
5	Waiting for teammate to hit ball	WTP
6	Waiting for opponent to hit ball	WOP
0	Background	BG

C.2 Trajectory Length Statistics

As the distance measurements in the datasets we used are different. NBA dataset measures distance in feet, while in tennis and dance datasets the distance is measured in pixels. Here we provide a statistic of trajectories distance in these three datasets Table 2 so that the performance among different length categories and different datasets can be better understood.

Table 2. Statistics of trajectory length in pixels of tennis dataset (resolution 1920×1080) and dance dataset (resolution 640×480).

Tennis (1920×1080)	Minimum Length	Maximum Length	Median Length
Short Trajectories	23.60	115.06	85.54
Middle Trajectories	115.30	230.02	159.93
Long Trajectories	230.50	524.00	279.86
NBA (100×50)	Minimum Length	Maximum Length	Median Length
Short Trajectories	0	5.99	3.16
Middle Trajectories	6.01	11.96	8.30
Long Trajectories	12.01	28.00	15.32
Dance (640×480)	Minimum Length	Maximum Length	Median Length
Short Trajectories	0	63.96	23.83
Middle Trajectories	64.27	127.95	94.31
Long Trajectories	128.13	422.17	169.68

D Additional Experimental Results

D.1 Ablation Study of Decoders

In the typical decoder, the query encompass past estimated target participants, while key & value are the observed participants in all past frames. “All Query” indicates the straightforward entry-flipping, where query is based on observed participants in all past frames, while key & values are the estimated target participants. “Limited Query” is what was presented in our main paper, in which only observed participants in the current frame is present in the query. Since two layers of decoder are applied, “Hybrid” means employing one typical decoder and one EF-Decoder sequentially, where “TP→EF” means apply typical decoder first and vice versa. The results in Table 3 show that “Limited Query” achieves much better performance than “All Query”. The reason is that most of observations in past frames are less important than current frame, such as tennis and basketball where both position and speed change rapidly. Both hybrid decoders cannot outperform “Limited Query” as every single typical decoder layer will accumulate more errors than our method. Hybrid versions can only introduce less error than typical transformer.

Table 3. Comparisons of different Decoders on Tennis dataset.

Decoder	**MAD**				**FAD**			
	Short	**Mid**	**Long**	**Avg**	**Short**	**Mid**	**Long**	**Avg**
Typical	20.14	33.09	50.70	31.33	35.85	52.55	71.57	49.67
All Query	20.80	33.05	46.86	30.92	38.25	55.50	70.14	51.71
Limited Query	**19.24**	**30.71**	**41.98**	**28.44**	**34.97**	**50.36**	**62.60**	**46.83**
Hybrid(TP→EF)	19.49	31.01	45.71	29.29	35.31	50.96	69.81	48.43
Hybrid(EF→TP)	19.52	31.53	43.84	29.24	35.53	52.62	70.57	49.43

D.2 Trajectory Inference and Prediction

Table 4. Comparisons of trajectory inference with baselines and SOTA methods on dance dataset.

Task	**Methods**	**MAD**				**FAD**			
		Short	**Mid**	**Long**	**Avg**	**Short**	**Mid**	**Long**	**Avg**
Inference	CNN Based	7.07	11.78	14.58	11.05	8.77	14.26	16.31	13.01
	RNN Based	7.89	13.30	16.91	12.59	9.76	15.17	17.78	14.13
	Transformer	6.97	11.76	13.51	10.66	8.89	14.38	15.26	12.74
	SR-LSTM [2]	8.72	14.08	19.39	13.95	9.80	15.50	18.66	14.54
	STAR [1]	7.94	13.83	20.93	14.11	9.15	14.81	21.20	14.94
	EF-Transformer	**6.81**	**9.62**	**11.82**	**9.36**	**7.86**	**10.61**	**12.42**	**10.24**
Prediction	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 [2]	9.50	15.67	22.48	15.76	11.56	18.16	21.82	17.05
	STAR [1]	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**

Table 4 shows the comparisons between the task of trajectory inference and prediction on the dance dataset. It can be observed that performances of inference and prediction are similar in the dance dataset, which is different from results in the tennis dataset. Behavior of all dancers in a group dance have predefined patterns. Therefore, the information of future frames of observed dancers are not as important as in tennis dataset.

Table 5. Comparisons with baselines and SOTA methods on pedestrian dataset.

Methods	**Performance MAD/FAD**					
	ETH	HOTEL	ZARA	ZARA2	UNIV	AVG
SR-LSTM [2]	2.98/5.04	2.68/**4.70**	2.01/3.34	1.84/3.48	2.13/3.85	2.33/4.10
STAR [1]	4.04/6.12	3.87/5.99	4.69/8.44	4.05/7.11	4.81/9.02	4.29/7.34
Transformer	2.52/4.54	2.30/3.76	1.75/3.01	1.85/3.55	2.23/4.35	2.13/3.84
EF-Transformer	**2.30/4.32**	**2.66**/4.71	**1.50/2.58**	**1.48/2.75**	**2.07/3.63**	**2.00/3.60**

Table 5 shows the trajectory prediction results on the pedestrian dataset, where only the first frame of ground truth is provided for target participants. Although our EF-Transformer outperformed all compared methods except FAD on 'hotel', both MAD and FAD for all methods are significantly larger than results with 8-frame setting. As walking pattern of pedestrians highly relies on self intention information, which underlies the historical trajectories, it is difficult to do the prediction without history information. It is also can be observed in Fig. 5 that if observed pedestrians are irrelevant to the target pedestrian, results of all methods with 1-frame setting are likely to fail, *e.g.* , image at row 2 column 4.

D.3 Multi-Task Inference and Prediction

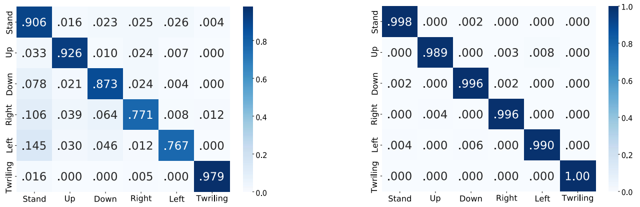
The results of multi-task inference and prediction on the tennis dataset is shown in Table 6, where a typical transformer is compared. As defined in Section C.1, action labels of different participants in one frame is complementary. For example in the serving stage, if the actions of observed participants are 'S' and 'WOS', then the action of the target participant will be 'WTS'. The action of the target participant can easily be deduced from seeing the actions of other observed participants, so both the typical transformer and our EF-Transformer achieved 100% accuracy for action inference and prediction. For our EF-Transformer, the trajectory inference and prediction results are also comparable between providing action ground truth and inferring actions simultaneously. The confusion matrices of two methods for action prediction is shown in Figure 1.

D.4 Robustness with Multiple Noise

We follow the setting of Section 4.6 in our paper and evaluate the performances of our method and typical transformer with multiple-frame noise involved. The results are shown in Table 7.

Table 6. Comparisons of multi-task inference and prediction with typical transformer on tennis dataset.'Traj' represents the task of trajectory inference or prediction, during which ground truth action labels are provided. 'Multi' represents the task of multi-task inference or prediction, where both trajectories and action labels have to be estimated.

	Inference	**MAD**				**FAD**			
		Short	**Mid**	**Long**	**Avg**	**Short**	**Mid**	**Long**	**Avg**
Traj	Transformer	21.17	32.91	46.67	30.95	37.14	52.06	68.14	49.34
	EF-Transformer	19.40	**30.04**	43.04	**28.35**	35.38	**48.62**	64.23	**46.43**
Multi	Transformer	20.22	33.42	49.76	31.36	36.11	52.65	72.79	50.02
	EF-Transformer	**19.21**	31.31	**42.85**	28.86	**33.62**	50.86	**63.88**	46.80
	Prediction	**MAD**				**FAD**			
		Short	**Mid**	**Long**	**Avg**	**Short**	**Mid**	**Long**	**Avg**
Traj	Transformer	20.14	33.09	50.70	31.33	35.85	52.55	71.57	49.67
	EF-Transformer	**19.24**	30.71	**41.98**	**28.44**	34.97	50.36	**62.60**	46.83
Multi	Transformer	20.32	33.71	50.81	31.71	36.50	55.10	76.96	52.01
	EF-Transformer	19.26	**30.14**	43.83	28.49	**33.27**	**48.30**	63.82	**45.45**



	Stand	Up	Down	Right	Left	Twirling
Stand	.906	.016	.023	.025	.026	.004
Up	.033	.926	.010	.024	.007	.000
Down	.078	.021	.873	.024	.004	.000
Right	.106	.039	.064	.771	.008	.012
Left	.145	.030	.046	.012	.767	.000
Twirling	.016	.000	.000	.005	.000	.979

(a) Transformer

	Stand	Up	Down	Right	Left	Twirling
Stand	.998	.000	.002	.000	.000	.000
Up	.000	.989	.000	.003	.008	.000
Down	.002	.000	.996	.002	.000	.000
Right	.000	.004	.000	.996	.000	.000
Left	.004	.000	.006	.000	.990	.000
Twirling	.000	.000	.000	.000	.000	1.00

(b) EF-Transformer

Fig. 1. Confusion matrices of action prediction on dance dataset.

E Additional Visualizations

We also provide additional visualizations of results on different datasets.

Fig. 2 shows some trajectory predictions on tennis dataset. Compared to typical transformer, the advantages of our EF-Transformer is significant especially for long and not smooth trajectories (rows 3 and 4 in Fig. 2). Some failure cases are shown and discussed in Fig. 3 Some videos are also provided to dynamically illustrate the trajectory prediction. In the videos, boxes show the positions of participants in the current frame, for which the red box is the target participant and white boxes are observed participants. Similarly, the red trajectory is the ground truth for the target participant and the white trajectories are for the observed participants. The trajectory predicted by our EF-Transformer is in green, and the prediction of the current frame is represented by a yellow arrow. The trajectory predicted by the typical transformer is in blue, and the prediction of the current frame is a cyan arrow. Note that when predicting the trajectory of the target participant in the current frame, the provided input information comprises white observed trajectories, the predicted target trajectory in past

Table 7. Comparisons of FAD on tennis dataset with multiple 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=2,3	95.66	122.22	162.25	119.27	39.88	65.46	99.97	61.95
Noise@t=5,6	75.86	101.57	141.87	98.97	50.62	65.48	94.78	64.97
Noise@t=8,9	134.23	164.20	211.25	161.18	132.16	143.73	170.83	144.05

frames (blue or green), the coarse ball trajectory, and action labels, while the red ground truth target trajectory is hidden and only shown for comparison.

Some results of NBA dataset are visualized in videos, where we can see that both of our EF-Transformer and typical transformer achieve good prediction performance. Some failure cases happen when the target player go to defend another opponent, which sometimes is also reasonable in real basketball games.

Results of trajectory prediction of two target dancers on dance dataset are shown in Fig. 4, from which we can observe that our method is capable of predicting more precise moving trends and patterns than the typical transformer.

Fig. 5 shows some results of trajectory prediction on pedestrian datasets, where each row of images are the examples of subset ‘eth’, ‘hotel’, ‘zara1’, ‘zara2’, and ‘univ’ respectively. The first two columns are results with the 8-frame setting, *i.e.* , 8-frame ground truth trajectory is provided for the target pedestrian. The last two columns are the results of same testing samples with the 1-frame setting. Qualitatively, our EF-Transformer provided best predictions on the 8-frame setting. For the 1-frame setting, our method can also predict the trajectory better than other methods when the selected observed pedestrians are relevant to the target pedestrian.

References

1. Yu, C., Ma, X., Ren, J., Zhao, H., Yi, S.: Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In: ECCV (2020)
2. Zhang, P., Ouyang, W., Zhang, P., Xue, J., Zheng, N.: Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In: CVPR (2019)

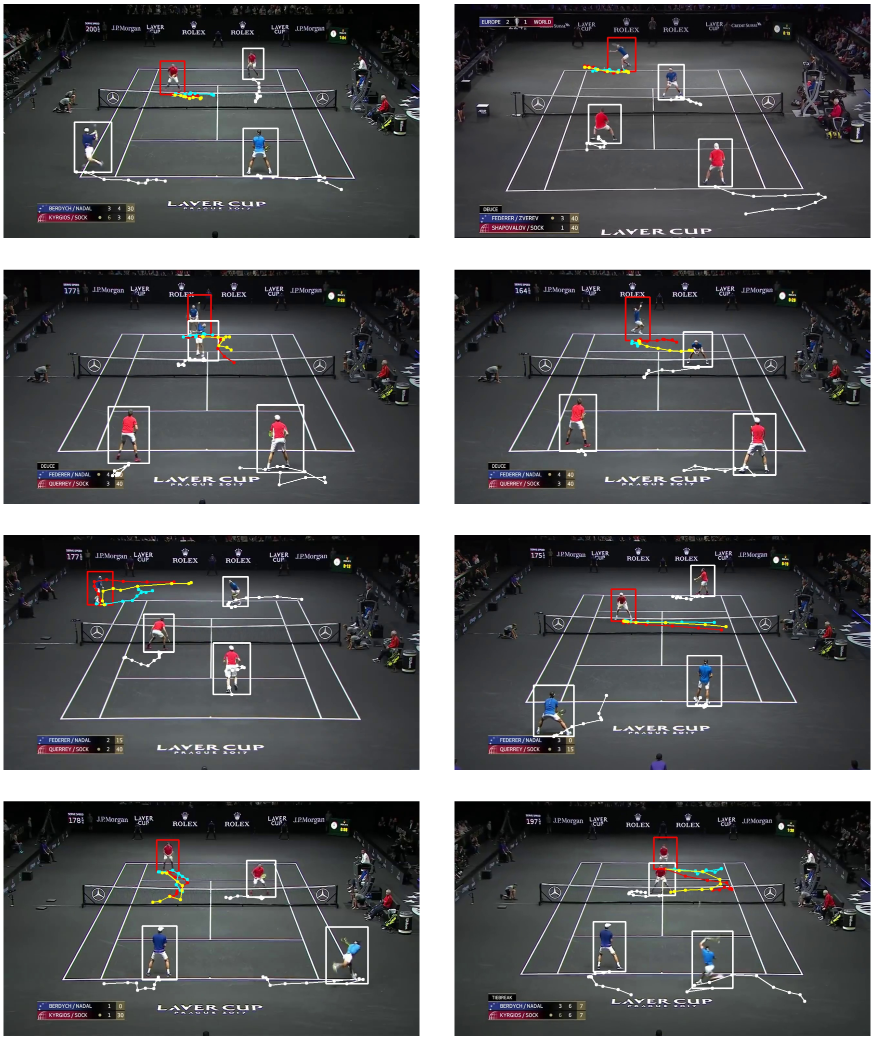


Fig. 2. Visualization of trajectory prediction results of EF-Transformer and typical transformer on tennis dataset. White rectangles and trajectories are the observed participants. Red rectangles are target participants with red trajectories for ground truth. Cyan trajectories are predicted by typical transformer and yellow ones are predicted by our method.

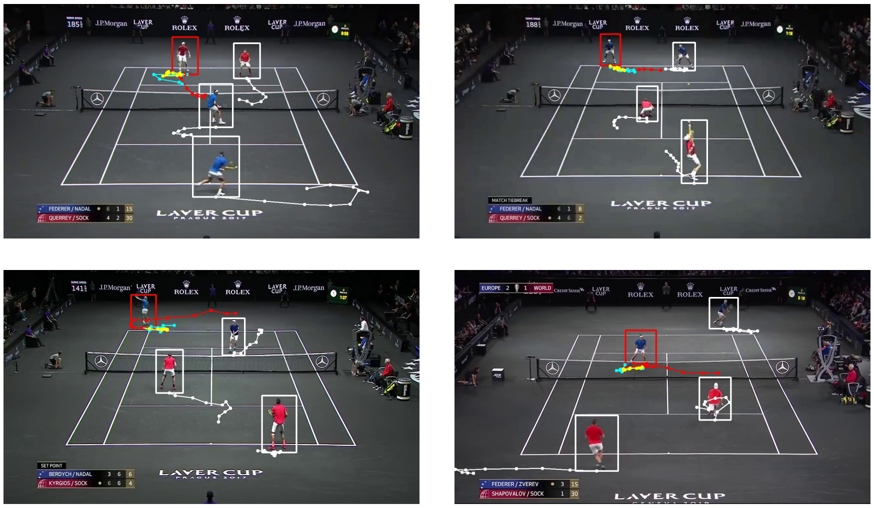


Fig. 3. Some failure cases of trajectory prediction in tennis dataset. White rectangles and trajectories are the observed participants. Red rectangles are target participants with red trajectories for ground truth. Cyan trajectories are predicted by typical transformer and yellow ones are predicted by our method.
The top two figures show that our model fails when the target player go towards his teammate as usually two players try to defense as much area as possible instead of stand close to each other.
The bottom left figure shows that our model suppose that the teammate of the target player will pursue the ball but in fact both of players try to get the ball in this round. The bottom right figure shows that our model choose to play safe and let his teammates to hit the ball back, however, the real player decide to intercept the ball close to the net to win this round.

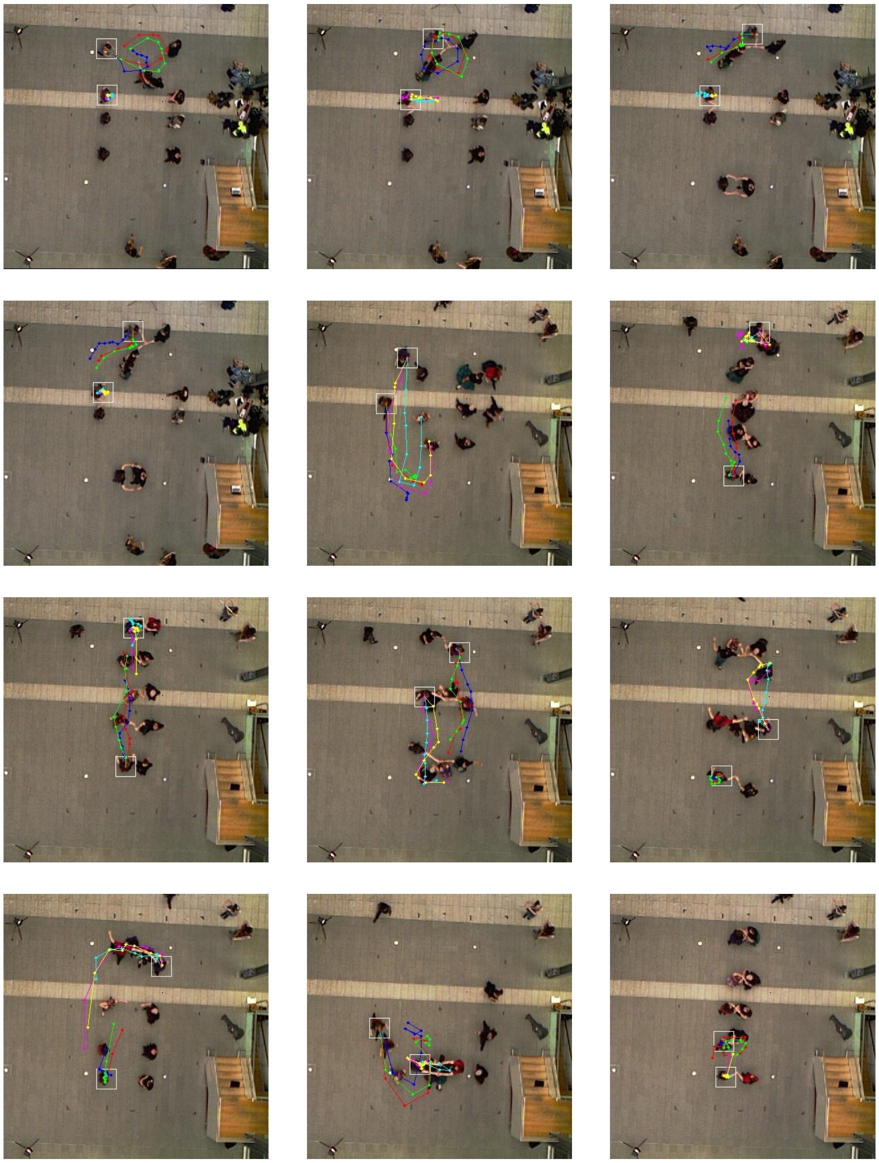


Fig. 4. Visualization of trajectory prediction results on dance datasets. White rectangles are initial positions of two target dancers. Ground truth trajectories are represented in red and magenta. Trajectories predicted by typical transformer are in blue and cyan, while by our EF-Transformer are in green and yellow. Frames are cropped to 400×400 for better view.

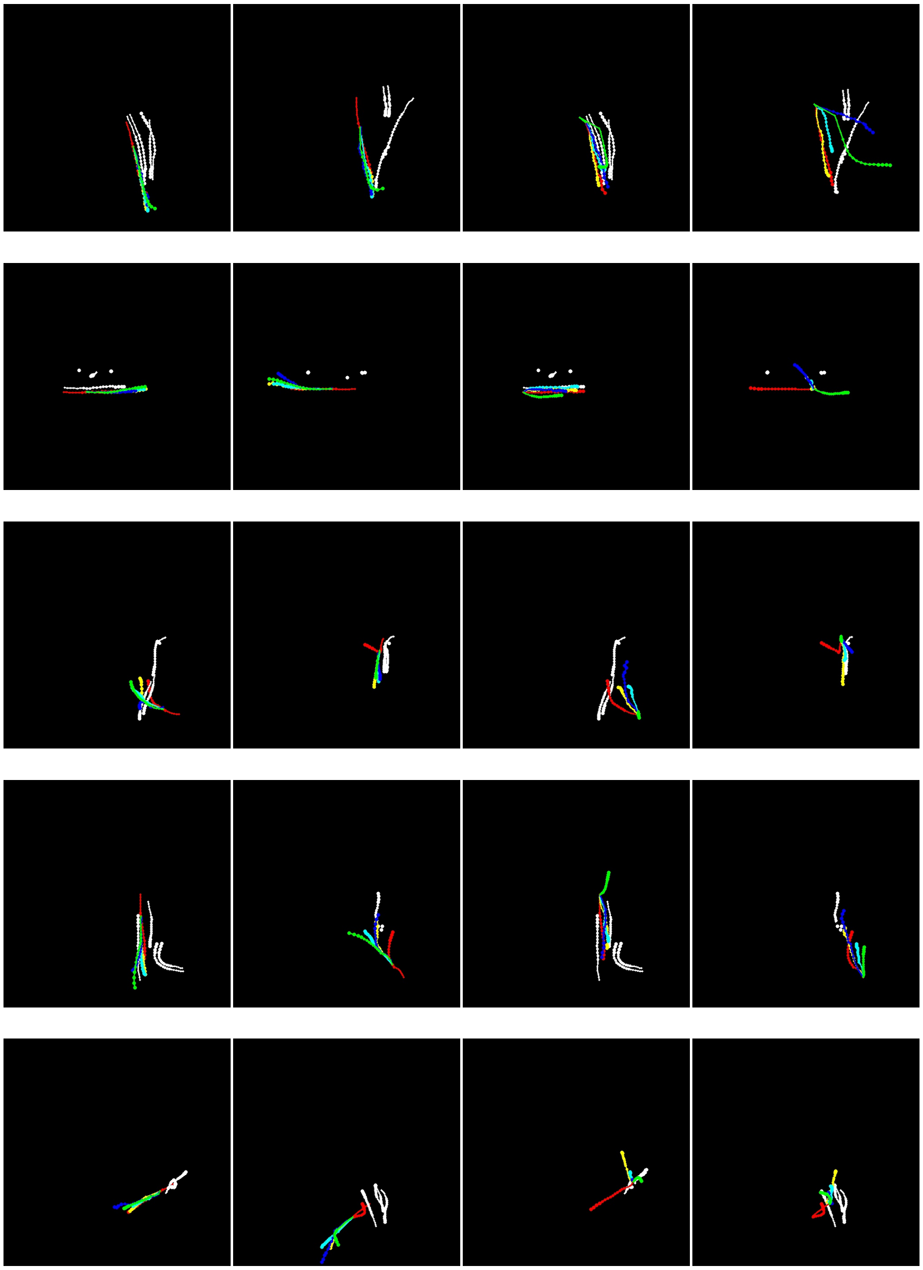


Fig. 5. Visualization of trajectory prediction results on pedestrian datasets. White trajectories are observed pedestrians, and red ones represent the ground truth of the target pedestrian. Yellow, cyan, blue, and green trajectories are predicted by our EF-Transformer, typical transformer, SR-LSTM [2] and STAR [1] correspondingly. The first two columns are results with 8-frame setting and last two columns are the results of same samples with 1-frame setting.