

Supplementary Material: GPRAR: Graph Convolutional Network based Pose Reconstruction and Action Recognition for Human Trajectory Prediction

BMVC 2021 Submission # 1035

In the main paper, we have presented GPRAR, a novel graph convolutional network based pose reconstruction and action recognition for human trajectory prediction. GPRAR consists of two novel sub-networks: PRAR (Pose Reconstruction and Action Recognition) and FA (Feature Aggregator). This material provides additional details of the network parameters we used in our experiments to achieve the reported results in the paper (Section 3 and 4). We also provide additional pose reconstruction results (Section 5) and trajectory prediction results with analysis (Section 6). The code will be made publicly available.

1 Details of Training Setup

Training Setup. Training is done in two stages: (1) PRAR plays a vital role in our prediction network; to be impactful, it is crucial to successfully train PRAR for pose reconstruction and action recognition before (2) customizing it for the prediction task. The details are as follows.

Stage 1: As TITAN and JAAD have limited numbers of complete human skeletons, we first train PRAR on Kinetics dataset [?] to obtain the initial network weights. Kinetics is a large human skeleton dataset, which contains around 300,000 video clips with 400 human action classes of various daily activities. Once PRAR is successfully trained on Kinetics dataset, we continue training the pre-trained PRAR on TITAN and JAAD datasets. The trained PRAR model on TITAN and JAAD datasets (obtained from the last training epoch) is used in the next training stage.

Stage 2: We customize the pre-trained PRAR to the trajectory prediction task on JAAD [?] and TITAN [?] datasets. Specifically, we attach FA module on top of PRAR and train the entire prediction model using the loss function in Equation ???. Our training setup is considered an adaptive learning approach as opposed to the non-adaptive learning where the input features to the predictor (i.e., FA in our work) are fixed. We show the effectiveness of this learning approach in the ablation study.

In both stages, our model is trained using stochastic gradient descent [?] with a learning rate of 0.01 and 50 epochs. We decay the learning rate by 0.1 after every 10 epochs. To implement spatial-temporal graph convolutions, we use similar implementation steps discussed in [?]. Our network model is implemented using PyTorch [?].

2 Pose Reconstruction and Action Recognition Results.

We found that pre-training PRAR on Kinetics dataset (in stage 1) significantly improves the pose reconstruction losses for both TITAN and JAAD datasets as depicted in Figure ???. Interestingly, by using the pre-trained network weights, pose reconstruction losses are significantly decreased (better) by 64% and 78% on TITAN and JAAD respectively in comparison with using random weights.

Figure ?? shows an example qualitative result of PRAR on JAAD validation data. We observe that PRAR is capable of reconstructing the missing human joints (e.g., missing head and legs in this figure). Quantitatively, we achieve pose reconstruction error of about 5 pixels and 10 pixels on TITAN and JAAD datasets given the image dimensions 1080×1080 pixels.

For the action recognition task, PRAR achieved 99% accuracy on JAAD with two action classes, and 91.05% on TITAN with eight action classes. We consider these to be desirable accuracies for skeleton-based action recognition.

3 Network Architecture of Pose Recognition and Action Recognition (PRAR).

PRAR consists of three main components: a pose encoder, a pose reconstruction decoder, and an action recognition decoder; each consists of multiple layers of spatial-temporal pose graph convolutional network (st-pgcn), as shown in Table 1. Each layer performs the spatial-temporal convolution (i.e., Equation 1 in the main paper) to estimate new values of each human joint. An important parameter of this convolution is the size of neighbor set $B(\cdot)$, the number of nearby spatiotemporally connected nodes. Similar to [?], we control the size of the neighbor set $B(\cdot)$ using a kernel with shape $[T, S]$, where T and S are the temporal and spatial sizes, respectively. To be specific, we use 3 layers for the encoder and 4 layers for each decoder. Each layer has temporal size of 9 and spatial size of 3. These numbers are selected based on our empirical study that yields the best saturated results reported in the paper.

The sequence of noisy observed human skeletons is denoted as K_{obs}^i with shape $[B, C, T_{obs}, \mathcal{K}] = [1, 3, 10, 18]$, where $B, C, T_{obs}, \mathcal{K}$ are batch size, channel size, observed time, and number of human joints, respectively. We first normalize K_{obs}^i using BatchNorm1d layer [?]. The output of BatchNorm1d is forwarded to the pose encoder to learn an encoded pose feature for both pose reconstruction and action recognition tasks. Specifically, the pose encoder increases the channel size of the input pose feature (i.e., $3 \rightarrow 64 \rightarrow 128 \rightarrow 256$), while we decrease the channel sizes of the encoded feature in both decoder branches. At the last layer of action recognition decoder, we use a fully connected network (fc) to convert the output pose feature of layer reg.st-pgcn4 to the appropriate size (i.e. $[1, \mathcal{N}]$) for action prediction. During the adaptive training process, this layer is customized to support a different number of action classes for each dataset (e.g., 400 action classes for Kinetics, 9 action classes for TITAN, and 2 action classes for JAAD).

4 Network Architecture of Feature Aggregator.

Table 2 shows the network architecture of Feature Aggregator, which consists of four feature encoders and a decoder. Each encoder encodes an input feature (i.e., reconstructed pose,

Layer Type	Kernel Shape $[\mathcal{T}, \mathcal{S}]$	Output Shape $[B, C, T_{obs}, \mathcal{K}]$
Input K_{obs}^i	-	[1, 3, 10, 18]
BatchNorm1d	-	[1, 3, 10, 18]
Pose Encoder		
enc.st-pgc1	[9,3]	[1, 64, 10, 18]
enc.st-pgc2	[9,3]	[1, 128, 10, 18]
enc.st-pgc3	[9,3]	[1, 256, 10, 18]
Pose Reconstruction Decoder		
rec.st-pgc1	[9,3]	[1, 256, 10, 18]
rec.st-pgc2	[9,3]	[1, 64, 10, 18]
rec.st-pgc3	[9,3]	[1, 32, 10, 18]
rec.st-pgc4	[9,3]	[1, 3, 10, 18]
Action Recognition Decoder		
reg.st-pgc1	[9,3]	[1, 256, 10, 18]
reg.st-pgc2	[9,3]	[1, 64, 10, 18]
reg.st-pgc3	[9,3]	[1, 32, 10, 18]
reg.st-pgc4	[9,3]	[1, 3, 10, 18]
reg.fcn	$[3, \mathcal{N}]$	$[1, \mathcal{N}]$

Table 1: PRAR Network Parameters. \mathcal{T} : temporal kernel size, \mathcal{S} : spatial kernel size. B : batch size, C : channel size, T_{obs} : observed time, \mathcal{K} : number of human joints. \mathcal{N} : number of action classes.

location, action, and camera motion) using multiple layers of one-dimensional temporal convolution (conv1d), rectifier linear unit (ReLU) [?], and batch normalization (BN) [?]. Next, the encoded features are channel-wise concatenated in the intermediate layer, which is used as an input to the decoder to produce the future trajectory.

5 Additional Trajectory Prediction Results

We present additional pose reconstruction results (Figure 1) in various extreme scenarios where a pose detector fails to detect, but PRAR successfully reconstructs. These scenarios include: a pedestrian occluded by another pedestrian (row 1) or by another object (rows 2 and 3); a small-scale pedestrian, who is far from the camera (row 4). We also show a failure case of PRAR, where the pedestrian is too far from the camera (last row). In this scenario, since the scale of this pedestrian is too small and the structure of skeleton is not maintained (i.e., the detected human joints are too close with each other), PRAR fails to utilize the structural information of human skeletons to reconstruct the pose.

6 Additional Pose Reconstruction Results

In this section, we first analyze the prediction performance of GPRAR with varying future prediction time (Section 6.1) and present additional qualitative prediction results (Sec-

tion 6.2).

6.1 Results with varying future prediction time.

We illustrate the prediction for future prediction steps by varying T_{pred} on JAAD dataset in Table 3. Given the observed time $T_{obs} = 10$ frames, we report the prediction errors (ADE/FDE in pixels) under noisy observations in different timesteps T_{pred} , ranging from 10 frames to 30 frames. For interpretability, 30 frames correspond to 3 seconds into future considering the frame per second (fps) is 10. We make comparisons with FPL method, which is closely related to our model. In general, the prediction errors of both methods increase when the prediction time increases. However, GPRAR still outperforms FPL significantly in all prediction steps. Especially, GPRAR shows prediction accuracy improvement by 16% relative to FPL at $T_{pred} = 30$.

6.2 Additional qualitative prediction results.

We present additional prediction results of GPRAR in different scenarios of noisy poses (Figure 2). With the success of PRAR in reconstructing these noisy poses, our prediction results (yellow) are very close to the ground truth trajectories (red). We show a failure case of GPRAR (Figure 2, last row), where a small-scale pedestrian detected, PRAR fails to completely reconstruct the human pose (corresponding to Figure 1 last row), thus leading to the failure of GPRAR.

We also provide additional prediction results for various human actions, as shown in Figure 3. We observe that GPRAR has successfully considered different action types into trajectory prediction. For example, a ‘running’ pedestrian (row 4, left image) moves faster (i.e. longer predicted trajectory) than “sitting”, “bending”, or “standing” pedestrians. We note that the pedestrians’ actions and movements must be considered relative to the camera motion. For example, a “sitting” pedestrian may have large motions (row 2, right image) when the camera moves fast. This camera motion has also been considered and incorporated in GPRAR.

Layer Type	Kernel Shape $[\mathcal{T}, \mathcal{S}]$	Output Shape $[B, C, T_{obs}, \mathcal{K}]$
Pose Encoder		
conv1d + ReLU + BN	3	[1, 54, 10, 1]
conv1d + ReLU + BN	3	[1, 32, 10, 1]
conv1d + ReLU + BN	3	[1, 64, 10, 1]
conv1d + ReLU + BN	3	[1, 128, 10, 1]
Location Encoder		
conv1d + ReLU + BN	3	[1, 2, 10, 1]
conv1d + ReLU + BN	3	[1, 32, 10, 1]
conv1d + ReLU + BN	3	[1, 64, 10, 1]
conv1d + ReLU + BN	3	[1, 128, 10, 1]
Action Encoder		
conv1d + ReLU + BN	3	[1, 54, 10, 1]
conv1d + ReLU + BN	3	[1, 32, 10, 1]
conv1d + ReLU + BN	3	[1, 64, 10, 1]
conv1d + ReLU + BN	3	[1, 128, 10, 1]
Camera Motion Encoder		
conv1d + ReLU + BN	3	[1, 24, 10, 1]
conv1d + ReLU + BN	3	[1, 32, 10, 1]
conv1d + ReLU + BN	3	[1, 64, 10, 1]
conv1d + ReLU + BN	3	[1, 128, 10, 1]
Intermediate Layer		
Concatenation	-	[1, 128×4 , 10, 1]
conv1d + ReLU + BN	1	[1, 128, 10, 1]
Camera Motion Encoder		
deconv1d + ReLU + BN	3	[1, 128, 10, 1]
deconv1d + ReLU + BN	3	[1, 64, 10, 1]
deconv1d + ReLU + BN	3	[1, 32, 10, 1]
deconv1d + ReLU + BN	1	[1, 2, 10, 1]

Table 2: FA Network Details

T_{pred}	10	14	18	22	26	30
FPL	26.24/27.83	23.22/28.16	49.66/51.35	46.50/54.97	44.06/55.25	36.08/49.00
GPRAR	18.62/21.26	22.57/27.52	27.90/34.50	25.09/34.98	31.22/41.35	30.82/45.00

Table 3: Prediction results (ADE/FDE) with different prediction timesteps on JAAD dataset.

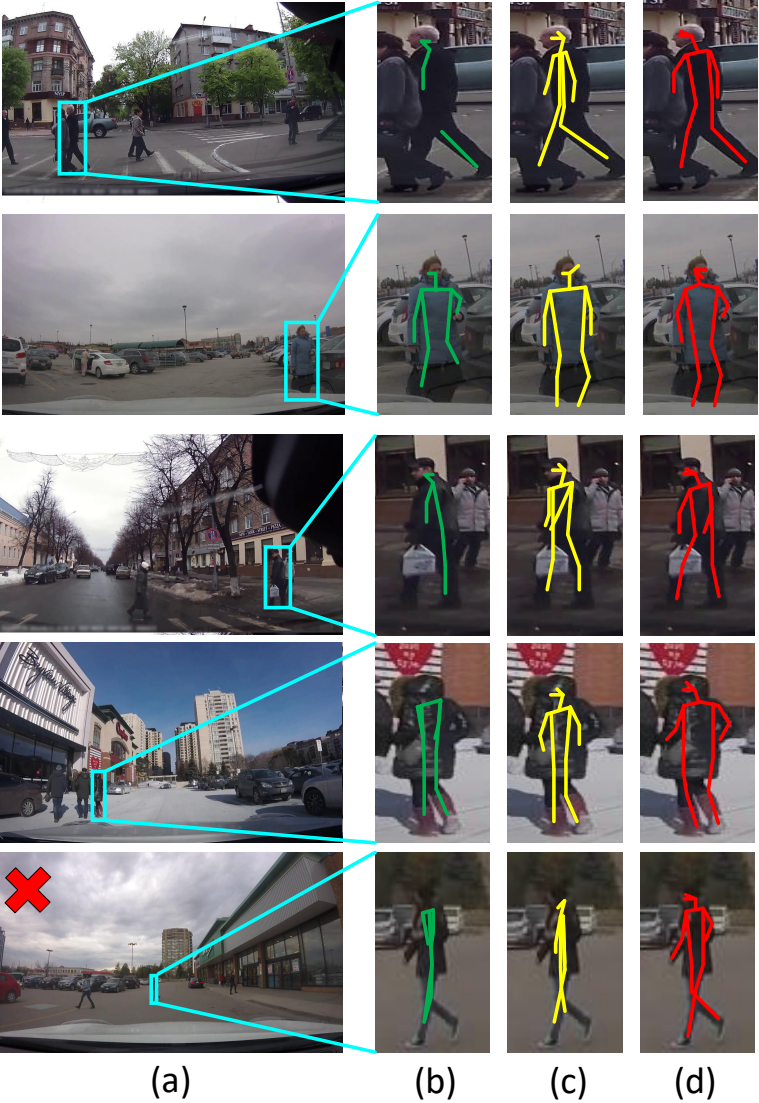


Figure 1: Additional Pose Reconstruction Results. (a) Original frame with a target pedestrian inside the bounding box, (b) noisy pose detection, (c) our reconstruction result, and (d) ground truth pose. A failure case is shown in the last row.

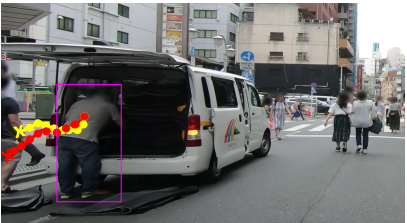


Ground truth trajectory GPRAR

Figure 2: Additional qualitative results of our model (GPRAR) in different noisy human poses. A failure case is shown in the last row.



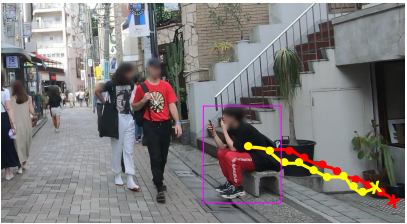
Bending



Bending



Sitting



Sitting



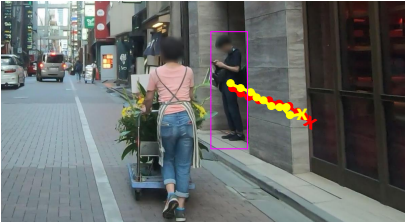
Running



Walking



Standing



Standing

Ground truth trajectory GPRAR

Figure 3: Additional qualitative results of our model (GPRAR) in different human actions.