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

Manh Huynh
http://cse.ucdenver.edu/~manhhuynh
Gita Alaghband
https://cse.ucdenver.edu/~gita

Department of Computer Science and Engineering
University of Colorado Denver
USA

**Abstract**

Prediction with high accuracy is essential for various applications such as autonomous driving. Existing prediction models are easily prone to errors in real-world settings where observations (e.g. human poses and locations) are often noisy. To address this problem, we introduce GPRAR, a graph convolutional network based pose reconstruction and action recognition for human trajectory prediction. The key idea of GPRAR is to generate robust features: human poses and actions, under noisy scenarios. To this end, we design GPRAR using two novel sub-networks: PRAR (Pose Reconstruction and Action Recognition) and FA (Feature Aggregator). PRAR aims to simultaneously reconstruct human poses and action features from the coherent and structural properties of human skeletons. It is a network of an encoder and two decoders, each of which comprises multiple layers of spatiotemporal graph convolutional networks. Moreover, we propose a Feature Aggregator (FA) to channel-wise aggregate the learned features: human poses, actions, locations, and camera motion using encoder-decoder based temporal convolutional neural networks to predict future locations. Extensive experiments on the commonly used datasets: JAAD [14] and TITAN [20] show accuracy improvements of GPRAR over state-of-the-art models. Specifically, GPRAR improves the prediction accuracy up to 22% and 50% under noisy observations on JAAD and TITAN datasets, respectively.

## 1 Introduction

Accurate prediction of human trajectory, i.e., forecasting pedestrians' future locations given their past (observed) frames in dynamic scenes, is critical for various applications such as autonomous driving [17], robotic navigation systems [19], and pedestrian tracking [22]. For the most part, challenges associated with predicting future trajectories are due to the presence of a multitude of features that may influence human future paths such as camera motion (egomotion), human shapes (pose), past locations, and human actions. More importantly, these features are often noisy due to environmental and scene impediments, occlusions for example. This problem has significantly degraded the performance of feature extractors, which in turn degrades the accuracy of the existing prediction models.

© 2021. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

Recent deep-learning-based methods[1, 5, 24, 28] have shown promising prediction results in 'perfect' settings, where ground truth (or complete) observations are given. Using ground truth observations helps model human motion more accurately and may improve the prediction accuracies.

However, the ground truth data is unavailable during test time. This limits the potential applicability of these methods in practice. Other methods [21, 29] rely on pre-processing techniques to denoise the observations in advance of testing. These approaches mainly focus on pre-processing (i.e. reconstructing or denoising) the human skeleton, an important feature for prediction. However, they are easily prone to errors under harsh conditions, such as fast camera motion and occlusions, especially in dynamic scenes. In this work, the following challenges are addressed: (1) reconstruction of human pose, which is a non-trivial task in computer vision. To the best of our knowledge, none of existing methods successfully reconstruct human skeletons in dynamic video sequences by exploiting the structural properties of human skeletons spatially and temporally. (2) the use of low-level human pose features to learn the higher-level action features. So far, the skeleton-based action features have not been considered for prediction tasks.

We design GPRAR to predict human future trajectory under noisy observations in dynamic video scenes by devising solutions to the above challenges. It consists of two novel sub-networks: (1) a human pose reconstruction and action recognition network (PRAR) and (2) an encoder-decoder based Feature Aggregator (FA), shown in Figure 1. The underlying idea of PRAR is to reconstruct human poses and learn action features simultaneously from the noisy pose detections. To best exploit the coherent and structural properties of human skeletons, PRAR is implemented with an encoder and two decoders, where each encoder and decoder is a multi-layer spatiotemporal graph convolutional network operating on the naturally connected human joints (or pose graph). Furthermore, we propose an encoder-decoder FA to channel-wise aggregate the learned features: reconstructed poses and locations, actions, and camera motion using temporal convolutional networks (TCNs). The aggregated feature is then used to output the future trajectory of a target pedestrian. In summary, the contributions of this paper are as follows: (1) We propose an efficient and robust human trajectory prediction network (GPRAR) under noisy observations (Section 3). GPRAR consists of two novel sub-networks: a human pose reconstruction and action recognition network (PRAR) and (2) an encoder-decoder based Feature Aggregator (FA).(2) We evaluate our model on two commonly used datasets: TITAN and JAAD, and show that our method outperforms other methods with a large margin under noisy scenarios (Section 4). We also conduct ablation studies to demonstrate the effectiveness of each system component.

## 2 Related Work

**Trajectory Prediction in Dynamic Scenes.** Most of the recent works in this research [1, 18, 20, 21, 28, 29] rely on methodologies such as recurrent neural networks (RNNs) [23], temporal convolutional networks (TCNs) [16], or their variants [7, 9], which aggregate various input features during an observation period to model the relative human motion to the camera view. For instance, graph structures [24, 28] can formulate interactions between agents (e.g. pedestrians, vehicles) using their past locations to predict the future trajectory. Pedestrians' shapes, scales, locations, and camera motion can be integrated using TCNs [29]. Recently, human action has also been used for trajectory prediction task. For example, Malla et al. [20] extracts action feature from the Two-Stream Inflated 3D ConvNet (I3D) and uses

it as an input to their prediction model. Liang et al. [18] designs a complex two-branch network to simultaneously predict human future activities and trajectories. While the above methods have shown promising prediction results, they rely on ground truth human locations and pose features that are not available in real-world settings.

**Neural Networks on Graphs.** Spatial-temporal graph convolutional neural networks (ST-GCN) is originally proposed by Kipf et al. [13], which extends the convolution operations from Convolutional Neural Network (CNN) to graph. ST-GCN and its variants have been widely used to model spatioltemporal features, which benefit several applications such as scene graph generation [8], point cloud classification and segmentation [15] , action recognition [4], and semantic segmentation [16]. For the task of trajectory prediction, ST-GCN [24, 28] is mainly used to model social interactions in static videos given the ground truth pedestrians' locations. Our work, instead, tackles the prediction problem in dynamic video sequences, where the social interactions become less effective given the dynamic changes of a front-view camera. Another model which is technically related to ours is skeleton-based ST-GCN [30]. However, this model is originally designed for action recognition task and assumes that complete human skeletons are available as inputs. To the best of our knowledge, GPRAR is the first prediction model that leverages graph convolutional network to simultaneously model robust human poses and action features for trajectory prediction under noisy observations.

# 3 System Design

**System Overview.** As illustrated in Figure 1, the task is to predict the future locations of $N$ pedestrians given the past $T_{obs}$ frames. For simplicity, let us describe this task for a target pedestrian $i \in N$ as follows:

**1.** At each time (current frame) $t_0$, our model receives noisy features extracted from the past $T_{obs}$ frames as inputs and produces the future trajectory $\hat{p}^i$ of pedestrian $i$ in the next $T_{pred}$ frames. We use two sequences of noisy input features: human skeletons $K^i_{obs}$ and optical flow $E_{obs}$ obtained using available public detectors (e.g., OpenPose [3] and FlowNet [10]). Our model is specially designed to deal with noisy or incomplete human skeletons generated as outputs of a feature detector on occluded pedestrians.

**2.** Given the sequence of observed noisy (incomplete) human skeletons $K^i_{obs}$ of pedestrian $i$, PRAR, a novel encoder-decoder based spatiotemporal pose graph convolutional network, reconstructs (denoises) these noisy human skeletons and recognizes the human actions. The sequence of reconstructed pose $\tilde{K}^i$ and action feature $A^i$ are then used as inputs to Feature Aggregator (FA) to predict the future locations. Moreover, since the observed location is an important feature to represent the overall human movements, we extract the sequence of reconstructed locations $\tilde{L}^i$ from the sequence of reconstructed pose $\tilde{K}^i$ and forward it to FA in the next stage. On the other branch (Figure 1), we calculate the grid optical flow $G$, by dividing each optical flow image $e_t \in E_{obs}$ into grids of $3 \times 4$ and averaging the values of all pixels in each grid cell. The grid optical flow represents the camera motion in different regions of the scene, thus provides more accurate camera motion compared to the pixel-level optical flow. In the last stage, Feature Aggregator (FA) aggregates all learned features $G, \tilde{K}^i, \tilde{L}^i, A^i$ via an encoder-decoder based one-dimensional temporal convolutional network to predict the future trajectory $\hat{p}^i$ of pedestrian $i$ in the next $T_{pred}$ frames. We consider independent movements among pedestrians, therefore, the same process above is applied to N pedestrians in the same observation time and predict N output trajectories via batch

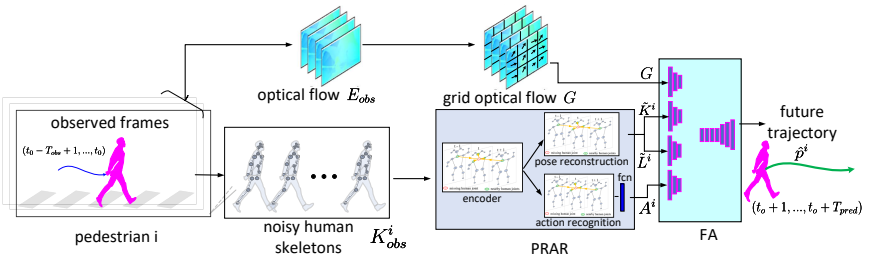


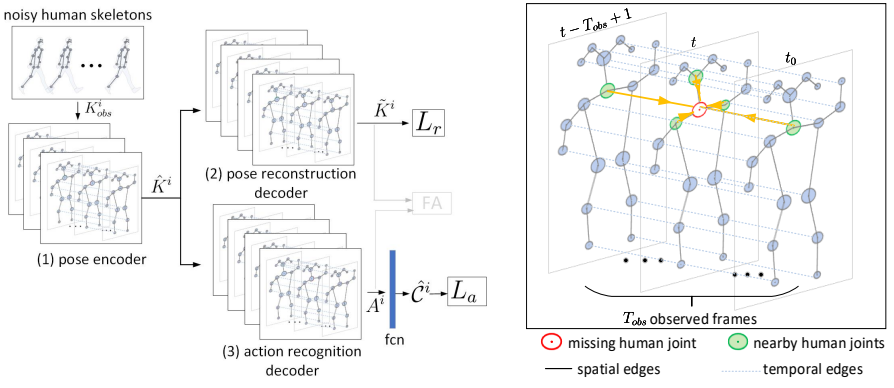

Figure 1: **GPRAR System Overview.** Our prediction model consists of two sub-networks: PRAR and FA. Given the sequence of observed noisy human skeletons $K^i_{obs}$ of the target pedestrian $i$ as input, PRAR reconstructs (denoises) the noisy human skeletons and recognizes the human action. In the later stage, FA aggregates the learned features: grid optical flow $G$, reconstructed human skeletons $\tilde{K}^i$, locations $\tilde{L}^i$, and action feature $A^i$ and predicts the future trajectory of pedestrian i.

processing.

**Notations.** We denote the sequence of observed skeletons of pedestrian $i$ as $K^i_{obs} = \{\mathbf{k}^i_t, \forall t = \{t_0 - T_{obs} + 1, \ldots, t_0\}\}$, where $\mathbf{k}^i_t = \{v^i_{kt}, \forall k = \{1 \ldots, \mathcal{K}\}\}$ is a human skeleton consisting of $\mathcal{K}$ human joints $v^i_{kt}$ of pedestrian $i$ at time $t$. The initial value of human joint $v^i_{kt}$ is $f(v^i_{kt}) = (x^i_{kt}, y^i_{kt}, c^i_{kt})$, where $(x^i_{kt}, y^i_{kt})$ is the two-dimensional coordinate and $c^i_{kt}$ is the confidence value of that joint. We note that not all human joints are visible due to occlusions or poor performance of the human detector. Thus, we set the initial value of missing human joints to $(0,0,0)$. The sequence of reconstructed human skeletons $\tilde{K}^i$ has the same size as $K^i_{obs}$. The sequence of observed optical flow is denoted as $E_{obs} = \{e_t, \forall t \in \{t_0 - T_{obs} + 1, \ldots, t_0\}\}$, where $e_t$ is the optical flow image with dimension $W \times H$ (width x height). We note that while the sequence of human skeletons $K^i_{obs}$ is unique for each pedestrian, the flow $E_{obs}$ is shared for all pedestrians during the same observation time. The sequence of reconstructed locations is written as $\tilde{L}^i = \{(\tilde{x}^i_t, \tilde{y}^i_t), \forall t = \{t_0 - T_{obs} + 1, \ldots, t_0\}\}$, where $(\tilde{x}^i_t, \tilde{y}^i_t)$ is the two-dimensional coordinate of the middle hip of human skeleton of pedestrian $i$ at time $t$. The learned action feature of pedestrian $i$ is denoted as $A^i = \{a^i_t, \forall t = \{t_0 - T_{obs} + 1, \ldots, t_0\}\}$, where $a^i_t$ is the action feature in each frame $t$. Lastly, the predicted trajectory of pedestrian i is denoted as $\hat{p}^i = \{(\hat{x}^i_t, \hat{y}^i_t), \forall t = \{t_0 + 1, \ldots, t_0 + T_{pred}\}$, where $(\hat{x}^i_t, \hat{y}^i_t)$ is the predicted location in the future frame. Next, we present the intuitions and design details of our prediction networks: PRAR (Section 3.1), FA (Section 3.2).

## 3.1 Pose Reconstruction and Action Recognition Network (PRAR)

The goal of PRAR is to generate robust action and pose features that boost the prediction under extreme scenarios (e.g., occlusions, fast camera motion, etc.) and those that compromise the susceptibility of a public detector. To this end, we design PRAR consisting of an encoder and two decoders, each of which comprises multi-layer of the graph neural networks (GNNs) operating on the sequence of observed human skeletons, shown in Figure 2. Before discussing the details, let us highlight PRAR with the following technical novelties: (1) while some works [6, 30] use GNNs on human skeletons solely for the task of action recognition, GNNs have not been utilized/extended for the pose reconstruction and human trajectory pre-




(a) (b)

Figure 2: **PRAR Network Architecture.** (a) PRAR consists of a pose encoder and two decoder branches for pose reconstruction and action recognition. (b) A single layer of each pose encoder/decoder is designed using the skeleton-based spatial-temporal graph convolutional network. An example of reconstructing the missing joint is also shown in (b), where it can be achieved by considering the information from nearby joints in both spatial and temporal domains.

diction tasks. (2) To the best of our knowledge, PRAR is the first encoder-decoder based GNNs for multi-task learning, in which robust pose features are learned to benefit both tasks mentioned above. (3) PRAR is a plug-and-play module that is trained separately but also can be integrated and jointly trained with other models for human trajectory prediction. We illustrate this training setup in Section 4.

As shown in Figure 2a, PRAR consists of three main components. (1) A pose encoder that takes the sequence of noisy observed human skeletons $K^i_{obs}$ as input and learns comprehensive encoded pose features $\hat{K}^i$. (2) A pose reconstruction decoder that takes the encoded pose features $\hat{K}^i$ and produces a complete (denoised) sequence of human skeletons $\tilde{K}^i$. (3) An action recognition decoder that also uses the shared pose feature $\hat{K}^i$ to generate action feature $A^i$, which is then input to a fully connected network (fcn) to generate an action class $\hat{C}^i$ of pedestrian $i$. While the learned features $\tilde{K}^i$ and $A^i$ are forwarded to FA module for trajectory prediction, $\tilde{K}^i$ and $\hat{C}^i$ are used to train PRAR separately at the pre-training phase using reconstruction loss $L_r$ and action recognition loss $L_a$. Since each encoder/decoder consists of multiple layers of GNNs, let us first present the details of a single layer $(l)$, illustrated in Figure 2b. In each layer, we model the sequence of observed human poses as a spatial-temporal skeleton-based graph $G = (V, E)$, where the nodes $V = \{v_{kt}, \forall t \in \{t_0 - T_{obs} + 1, \ldots, t_0\}, \forall k \in \{1, \ldots, \mathcal{K}\}\}$ are all the human joints in a skeleton sequence. The edge set $E$ consists of a spatial edge set $E_{sp} = \{v_{kt}v_{jt}, (k, j) \in \mathcal{K}\}$ that connects human joints naturally within a frame, and a temporal edge $E_{tp} = \{v_{kt}v_{k(t+1)}, \forall t \in \{t_0 - T_{obs} + 1, \ldots, t_0\}, \forall k \in \{1, \ldots, \mathcal{K}\}\}$ that connects the same human joint in consecutive observed frames. We note that Yan et al. [30] and Cheng et al. [6] utilize similar skeleton-based graph neural networks. However, these works do not apply to our case because they only focus on modeling human actions. PRAR is designed not only to learn a spatiotemporal action feature, but also to reconstruct the human missing joints. The reconstruction task is done through spatial-temporal graph convolutions by leveraging the message passing mechanism, in which directly-connected nodes can communicate with each other. Specifically, the spatial-temporal graph convolu-

tion is utilized to estimate the coordinate of a missing human joint using the information of nearby human joints in both spatial and temporal domains. In Figure 2b, we assume human neck as a missing human joint to illustrate this reconstruction process in details. The coordinate of a missing human joint at the observed frame $t \in \{t_0 - T_{obs} + 1, \ldots, t_0\}$ is calculated by gathering information from nearby visible human joints (nose, right shoulder, and left shoulder) within the same frame $t$ and information of the same joint from neighboring observed frames. In general, given the input feature of a missing node of pedestrian $i$ at layer $(l)$ is $f(v_{kt}^{i(l)})$, we apply the spatial-temporal graph convolution operation to estimate the new value $\tilde{f}(v_{kt}^{i(l)})$ of the node $v_{kt}^{i(l)}$ as:

$$\tilde{f}(v_{kt}^{i(l)}) = \sum_{v_{jt}^{i(l)} \in B(v_{kt}^{i(l)})} \frac{1}{z_t^{i(l)}} f(v_{jt}^{i(l)}) w_{jt}^{i(l)} \tag{1}$$

where $B(v_{kt}^{i(l)})$ is the set of nearby human joints $v_{jt}^{i(l)}$ that naturally connect to the node $v_{kt}^{i(l)}$ in both spatial and temporal axes; $w_{jt}^{i(l)}$ is a learnable weight vector; $z_t^{i(l)}$ is a normalization term, which normalizes the output features to range $[0,1]$. We note that the initial value of the missing node $v_{kt}^{i}$ at layer $(0)$ is $f(v_{kt}^{i(0)}) = (0,0,0)$, which is the noisy output of a human pose detector. The output feature $\tilde{f}(v_{kt}^{i(l)})$ at layer $(l)$ is forwarded to the next layer $(l+1)$ to calculate $\tilde{f}(v_{kt}^{i(l+1)})$. In the last layer of the pose reconstruction decoder, $\tilde{f}(v_{kt}^{i}) = (\tilde{x}_{kt}^{i}, \tilde{y}_{kt}^{i}, \tilde{c}_{kt}^{i})$ where $(\tilde{x}_{kt}^{i}, \tilde{y}_{kt}^{i})$ is the two-dimensional reconstructed location and $\tilde{c}_{kt}^{i}$ is the new confidence score. Note that Equation 1 can be applied for other non-missing nodes as well. Interestingly, we found that using the spatiotemporal graph convolution for all nodes improves the prediction results as it enhances the coherency of human skeletons.

**Encoder and Decoders.** Although the spatial-temporal graph convolution in a single layer is useful for reconstructing human poses, it only considers the information from the nearby nodes, which directly connect to the missing node. In fact, the other non-directly connected nodes may also have impacts on the missing nodes (e.g., given pedestrian's head locations, we, as humans, can estimate the locations of non-directly connected legs). Based on this intuition, we design each encoder and decoder with multiple layers of spatial-temporal graphs as multi-layers allow non-directly connected nodes to have impacts on the target node. Specifically, we use three layers for encoder and four layers for each decoder. These specific numbers of layers are determined based on our empirical study, in which we achieved saturated prediction results. We discuss the details of these network parameters in the supplementary materials.

As a result, the last layer of pose reconstruction decoder produces the reconstructed pose $\tilde{K}^i$. The last layer of action recognition decoder outputs the learned action feature $A^i$, which is then forwarded to fcn to calculate the action label $\hat{C}^i$. While the action label $\hat{C}^i$ is used to optimize PRAR for action recognition task, $A^i$ is used as input to FA in the next stage.

**Losses.** PRAR is initially trained separately from our prediction model (at pre-training phase) with the proposed multi-task loss as below:

$$L = L_r + L_a \tag{2}$$

$$L_r = \sum_{i}^{N} \sum_{k}^{K} \sum_{t}^{T_{obs}} \|\tilde{f}(v_{kt}^{i}) - \bar{f}(v_{kt}^{i})\|^2 \tag{3}$$

$$L_a = \sum_{i}^{N} ce(\hat{C}^i, \bar{C}^i) \tag{4}$$

where $L_r$ is pose reconstruction loss and $L_a$ is action recognition loss. For pose reconstruction loss, we use mean-square-error loss over predicted human joints with $\tilde{f}(v_{kt}^i)$ and $\bar{f}(v_{kt}^i)$ are reconstructed and ground truth values of human joint $v_{kt}^i$. uring training, PRAR is trained using sequences of complete poses sampled from training sets; creating new samples with missing joint obtained by randomly dropping the joints from the complete poses. For action recognition loss, we use the cross-entropy loss: where $ce$ is cross-entropy function; $\hat{C}^i$ and $\bar{C}^i$ are the predicted and ground truth action class labels of pedestrian $i$, respectively.

### 3.2 Feature Aggregator (FA)

The goal of FA is to channel-wise aggregate all the learned features: reconstructed pose $\tilde{K}^i$, reconstructed location $\tilde{L}^i$, action feature $A^i$, and regional optical flow $G$. To this end, we design an encoder-decoder based on temporal neural networks, where the encoder channel-wise aggregates these features, the decoder takes the aggregated (encoded) feature as input and generates the future trajectory $\hat{p}^i$. Specifically, FA consists of four feature encoders and a decoder. Each encoder encodes an input feature (i.e., reconstructed pose, location, action, and camera motion) using multiple layers of one-dimensional temporal convolution (conv1d), rectifier linear unit (ReLU) [25], and batch normalization (BN) [11]. 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 trajectories. The model details of FA are provided in the supplementary materials. Although Yagi et al. [29] (FPL) adopted a closely related network architecture, our FA is a much smaller network that outperforms FPL. Specifically, the number of network parameters of FA is 33% less than FPL's even though FA accommodates more input features. We believe this is due to the effectiveness of PRAR, which has produced the robust learned features. The model details of FA are provided in the supplementary materials. Once PRAR is trained separately for pose reconstruction and action recognition using loss function in Equation 2, the entire prediction network is trained using mean-square-error loss as $L(W) = \sum_i^N \sum_t^{T_{pred}} \|\hat{p}_t^i - \bar{p}_t^i\|^2$, in which $W$ includes all the trainable parameters of the model, $\hat{p}_t^i$ and $\bar{p}_t^i$ are the predicted and ground truth locations of pedestrian $i$ at time $t$, respectively.

## 4 Experiments

**Datasets.** We evaluate our model on two publicly available autonomous driving datasets: JAAD [14] and TITAN [20]. JAAD contains 346 videos with 82,032 frames. The videos are recorded at frame rate 30fps by a front-view wide-angle camera mounted in the center of the front windshield. The annotation rate is 30Hz. These videos range from short 5-second clips to 15-second videos shot under various scenes, weathers, and lighting conditions. TITAN consists of 700 video clips (60fps) annotated at 10Hz with 75,262 annotated frames and 395,770 persons. The annotations provide a variety of pedestrians' actions and interactions. We use 9 action classes (standing, jumping, squatting, bending, running, walking, laying down, sitting, kneeling) provided by TITAN, while JAAD supports two action classes: walking and standing. Note that action classes are unevenly distributed. We conduct experiments in each dataset separately by splitting the number of total videos in each

dataset into train/validation sets with ratio of 80/20. Similar to [29], we use 10 frames for observation and predict the next 10 frames (i.e., 0.3 second and predict next 0.3 second) for JAAD dataset. For TITAN datasets, we observe 10 annotated frames (1 second) and predict trajectories in next 20 annotated frames (2 seconds) as similar to [20].

**Training Setup.** Training is done in two stages as follows:

**Stage 1: Pre-training PRAR for pose reconstruction and action recognition**. As TITAN and JAAD have limited numbers of complete human skeletons and human actions, we first train PRAR on large-scale Kinetics dataset [12] to obtain our initial network weights, then fine-tune it on TITAN and JAAD datasets. We show the effectiveness of pre-training PRAR on Kinetics in the supplementary material. We use the same train/validation split as in [30] to train PRAR. To generate training data, we only extract complete pose samples and use them as ground-truth. During training, the noisy poses, inputs to PRAR, are obtained by randomly dropping a number of human nodes (i.e., set them to missed nodes) with occlusion rate from 0% to 50%. For validation, all pose samples are used. The action label, associated with each pose provided by Kinetics dataset, are also used to train PRAR with loss function in Equation 2. The model with best validation result is continued to train (fine-tune) on JAAD and TITAN datasets. We used the same method above to extract pose samples for JAAD/TITAN datasets. The best validated models are used in next training stage.

**Stage 2: Training GPRAR for trajectory prediction**. We customize the pre-trained PRAR to the trajectory prediction task on JAAD [14] and TITAN [20] datasets. Specifically, we attach FA module on top of PRAR and train the entire prediction model on the training set of each dataset. While training the entire GPRAR, we allow PRAR's network parameters to be updated. This is the adaptive learning approach. We show its effectiveness over non-adaptive one in the ablation study. In both stages, our model is trained using stochastic gradient descent [2] with a learning rate of 0.01 and 50 epochs. We decay the learning rate by 0.1 after every 10 epochs. The model obtained at the last training epoch is used to validated on the validation set. To implement spatial-temporal graph convolutions, we use similar implementation steps discussed in [30]. Our network model is implemented using PyTorch [26].

**Comparison Methods.** We compare our prediction results with two baselines (Const-vel [27], LSTM [1]) and three state-of-the-art methods (Social-STGCNN [24], FPL [29], and TITAN [20]). (1) Const-Vel uses the velocity, calculated using the last two observed locations, to interpolate future positions linearly. (2) LSTM models individual pedestrian behaviors using LSTM cells. (3) Social-STGCNN models social interactions between pedestrians in the scenes by using graph neural networks. (4) FPL incorporates pose, scales, egomotion features using temporal neural networks. (5) TITAN uses action features in combination data from IMU sensors for predicting future locations.

**Evaluation Metrics.** We evaluate our system using two commonly used metrics [1, 29]: (a) average displacement error (ADE): mean square error over all locations of predicted and true trajectories; (b) final displacement error (FDE): mean square error at the final predicted and true locations of all human trajectories.

## 4.1 Quantitative Prediction Results on JAAD and TITAN Datasets.

Table 1 shows quantitative results on JAAD and TITAN datasets (results from training stage 2). We compare our method (GPRAR) with other methods in three different observation modes: noisy, pre-processed, and ground truth. In the noisy (raw) mode, the observed data (poses and locations) are the outputs of a pose detector. In the pre-processed mode, the data

| Models | JAAD | | | TITAN | | |
|---|---|---|---|---|---|---|
| | N. | Pr. | C. | N. | Pr. | C. |
| Const-vel [27] | 48.07/75.03 | 38.56/59.98 | 18.68/30.15 | 150.31/238.70 | 84.18/35.47 | 23.81/42.72 |
| LSTM [1] | 46.96/57.28 | 45.70/56.12 | 51.04/61.42 | 51.52/80.59 | 50.99/79.90 | 28.92/50.51 |
| Social-STGCNN [28] | 80.46/71.38 | 80.46/71.38 | 79.14/58.73 | 72.32/52.86 | 73.62/54.84 | 68.78/51.68 |
| FPL [29] | 27.07/26.92 | 28.01/31.22 | 24.85/27.31 | 52.26/80.93 | 34.13/50.92 | 14.09/19.98 |
| TITAN [20]* | - | - | - | - | - | **11.32/19.53** |
| GPRAR (Ours) | **21.09/21.62** | **18.13/20.88** | **14.79/20.38** | **26.17/38.58** | **24.49/34.85** | 12.56/20.36 |

Table 1: Quantitative results (ADE/FDE in pixels) on JAAD and TITAN datasets in different observation modes: noisy (N.), pre-processed (Pr.), and complete (C.). The lower the better. (*) Since the implementation is not publicly available, we use the results reported in [20].

are estimated using KNN-imputer as used in FPL [29]. The ground truth observation is the complete pose data with no missing joints. Our model outperforms other methods on JAAD dataset in all three different scenarios. Specifically, our prediction results are 50% and 22% better than FPL in the noisy mode on TITAN and JAAD datasets, respectively. In the ground truth mode, our model outperforms others on JAAD dataset and produces very close results to TITAN. However, we note that TITAN method uses the IMU sensor data as additional features for prediction, while our method only relies on image data.

## 4.2 Qualitative Results.

Figure 3 shows sample qualitative results comparing GPRAR with FPL and LSTM in various scenarios of noisy poses (Figure 3, top row) and action types (Figure 3, bottom row). Due to the pose reconstruction capability, GPRAR outperforms others under occlusion scenarios, such as: (a) the pedestrian is too close to the car; thus, there are missing head and ankles; (b) the pedestrian walks across the street with hidden right hand (c) the pedestrian walks away with missing face and ankles (not detected). As GPRAR also considers the skeleton-based action feature, GPRAR outperforms others in different human action scenarios: (d) walking across street, (e) standing and waiting, and (f) bending while unloading packages.

**Ablation study.** The importance of individual features in GPRAR model is shown in Figure 4a. We observe that using reconstructed location (XR) improves the prediction accuracy in comparison with using noisy location (X). Moreover, using reconstructed location in combination with other features: action (A), camera motion (C) and reconstructed pose (PR) further improves the results. Our full model, which aggregates all features, achieves the best prediction results.

**Impact of occlusions.** To observe the impact of occlusions on performance of GPRAR, we compare three different combination of features: (a) noisy location and noisy pose (X + P); (b) noisy location, noisy pose, and camera motion (X + P + C); (c) reconstructed location, reconstructed pose, and camera motion (XR + PR + C). As shown in Figure 4b, all three variants perform well when the occlusion ratio is low. However, the performances of both (X + P) and (X + P + C) are significantly degraded under high occlusion scenarios, while the proposed method with reconstructed pose (XR + PR + C) is less impacted by the occlusions and shows a steady good result (low ADE). Specifically, (XR + PR + C) reduces the prediction error (ADE) by 80% and 25% compared to (X + P + C) and (X + P) at the occlusion ratio of 50% (i.e., a half human body is occluded).

**Impact of adaptive learning.** Empirically, we found that the adaptive learning approach effectively improves the prediction accuracy. Figure 4c shows the effectiveness of this adaptive approach during training. It produces a lower stable loss in comparison with the non-adaptive

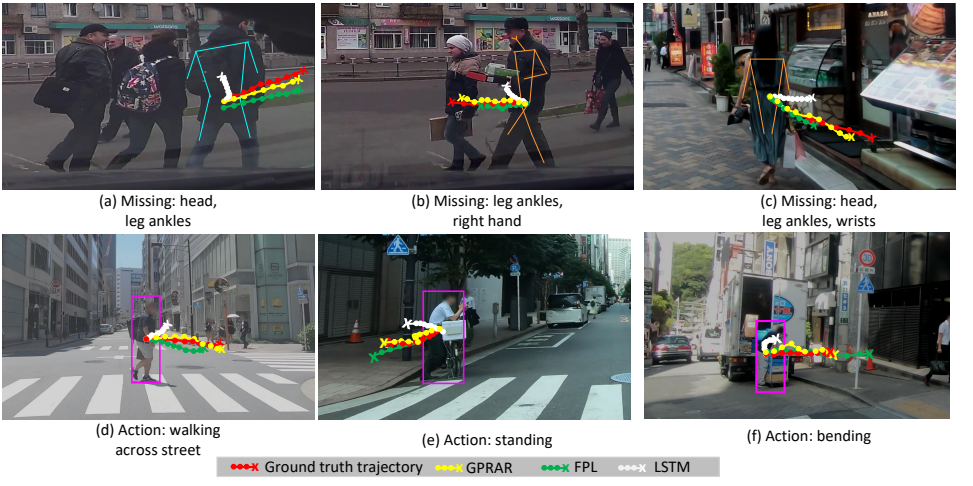
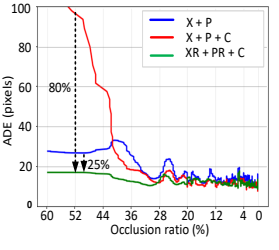
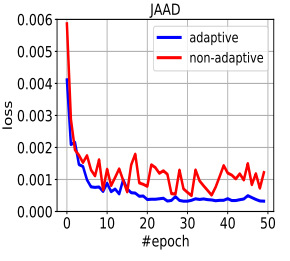



Figure 3: Qualitative results of our model (GPRAR) in comparisons with other models: FPL, LSTM for scenarios of missing human joints (top row) and action types (bottom row). Additional results are provided in the supplementary materials.

| Models | ADE | FDE |
|---|---|---|
| X | 30.55 | 45.42 |
| XR | 28.94 | 44.03 |
| XR + C | 27.92 | 41.58 |
| XR + A | 28.08 | 41.23 |
| XR + PR | 28.13 | 41.94 |
| **Full Model** | | |
| (XR + C + A +PR) | 26.17 | 38.58 |




(a) (b) (c)

Figure 4: Analysis: (a) Effects of each feature used (b) Impact of human occlusions; (c) effectiveness of our adaptive learning

one.

# 5 Conclusions

In this paper, we present GPRAR, a novel human future trajectory prediction model in dynamic video sequences, which efficiently handles noisy real-world scenarios. The main contributions of this paper are two novel subnetworks: PRAR and FA. While PRAR is trained for multi-task learning: action recognition and pose recognition, FA aggregates multiple learned features for trajectory prediction. The key implementation of PRAR is using the encoder-decoder based graph convolutional neural networks, which help exploit the structural properties of human poses. Through extensive experiments, we have shown GPRAR produces superior performance in comparisons with the state-of-the-art models. We have further presented performance analysis of introduced features and shown that our model performs effectively under occlusions and various human actions.

# References


[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016.

[2] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

[3] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[5] Rohan Chandra, Tianrui Guan, Srujan Panuganti, Trisha Mittal, Uttaran Bhattacharya, Aniket Bera, and Dinesh Manocha. Forecasting trajectory and behavior of road-agents using spectral clustering in graph-lstms. *IEEE Robotics and Automation Letters*, 5(3): 4882–4890, 2020.

[6] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 183–192, 2020.

[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[8] Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. Scene graph generation with external knowledge and image reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1969–1978, 2019.

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

[10] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[12] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[13] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[14] Iuliia Kotseruba, Amir Rasouli, and John K Tsotsos. Joint attention in autonomous driving (jaad). *arXiv preprint arXiv:1609.04741*, 2016.

[15] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4558–4567, 2018.

[16] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017.

[17] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, et al. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 163–168. IEEE, 2011.

[18] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G Hauptmann, and Li Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5725–5734, 2019.

[19] Manh Luong and Cuong Pham. Incremental learning for autonomous navigation of mobile robots based on deep reinforcement learning. *Journal of Intelligent & Robotic Systems*, 101(1):1–11, 2021.

[20] Srikanth Malla, Behzad Dariush, and Chiho Choi. Titan: Future forecast using action priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11186–11196, 2020.

[21] Karttikeya Mangalam, Ehsan Adeli, Kuan-Hui Lee, Adrien Gaidon, and Juan Carlos Niebles. Disentangling human dynamics for pedestrian locomotion forecasting with noisy supervision. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2784–2793, 2020.

[22] Huynh Manh and Gita Alaghband. Spatiotemporal ksvd dictionary learning for online multi-target tracking. In *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 150–157. IEEE, 2018.

[23] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

[24] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14424–14432, 2020.

[25] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[27] Christoph Schöller, Vincent Aravantinos, Florian Lay, and Alois Knoll. The simpler the better: Constant velocity for pedestrian motion prediction. *arXiv preprint arXiv:1903.07933*, 5(6):7, 2019.

[28] Chengxin Wang, Shaofeng Cai, and Gary Tan. Graphtcn: Spatio-temporal interaction modeling for human trajectory prediction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3450–3459, 2021.

[29] Takuma Yagi, Karttikeya Mangalam, Ryo Yonetani, and Yoichi Sato. Future person localization in first-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7593–7602, 2018.

[30] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.