Which Way Are You Going? Imitative Decision Learning for Path Forecasting in Dynamic Scenes – Supplementary Material

Yuke Li
York University, Toronto, Canada
AutoNavi, Alibaba Group, Beijing, China
ykleewh@yorku.ca

1. Motion Feature Construction

We propose to build a set of motion features upon displacement vectors to characterize the pedestrian trajectories in a dynamic scene. Let $\{x_t^i, y_t^i\}(t \in \{t_1, t_2, ...t_k\})$ be the notations of the coordinates of person i from t_1 to t_k. We can define the displacement for person i at time instance t by $\{x_{t_k}^i - x_t^i, y_{t_k}^i - y_t^i\}(t \in \{t_1, t_2, ...t_k\})$.

Fig.1 illustrates a detailed example of how to construct the 3D motion features $\mathcal{X}_t$ by means of these displacement vectors. We map the displacement of person i into $\mathcal{X}_t$ as following: $\mathcal{X}_t(x_{t_k}^i, y_{t_k}^i, 1)$ accommodates $x_{t_k}^i - x_t^i$, while $\mathcal{X}_t(x_{t_k}^i, y_{t_k}^i, 2)$ accommodates $y_{t_k}^i - y_t^i$. Formally, $\mathcal{X}_t$ at time instance t is defined as:

$$\begin{cases} \mathcal{X}_t(x_{t_k}^i, y_{t_k}^i, 1) = x_{t_k}^i - x_t^i + W \\ \mathcal{X}_t(x_{t_k}^i, y_{t_k}^i, 2) = y_{t_k}^i - y_t^i + H \end{cases} \tag{1}$$

Where H and W are the height and width of the input scene, respectively. We add W and H to Eq.1 to ensure all the entries of $\mathcal{X}_t$ are positive. In practice, $\{x_t^i, y_t^i\}$ are coordinates that have been projected to a dimension of $H \times W = 256 \times 256$ from annotations. Therefore, $\mathcal{X}_t$ have a dimension of $2 \times H \times W$. We include all individuals in the scene at time instance t through a single $\mathcal{X}_t$, owing to each non-zero index in $\mathcal{X}_t$ can be traced to a particular person. Our IDL is able to forecast the future paths of all moving objects in the scene simultaneously via $\mathcal{X}_t$.

The proposed motion features are built on the inputs of Behavior CNN[2]. We further make a significant contribution on forecasting multimodal future paths, in contrast of predicting a deterministic future as Behavior CNN does. Moreover, our work differs fundamentally from Behavior CNN in the network structure aspect.

In this study, we are interested in forecasting $\mathcal{X}_{t'}(t' \in \{t_{k+1}, t_{k+2}, ...t_{k+k'}\})$ by observing $\mathcal{X}_t(t \in \{t_1, t_2, ...t_k\})$. We filter the final predictions $\mathcal{X}_{t'}$ for both training and testing by computing $\mathcal{X}_{t'} = \mathcal{X}_{t'} \odot 1_{\mathcal{X}_{t'}, \mathcal{GT}_{t'}}$. $1_{\mathcal{X}_{t'}, \mathcal{GT}_{t'}}$ is an indicator function. This indicator function equals to one if $\mathcal{X}_{t'}$ have the same non-zero indexes with $\mathcal{GT}_{t'}$; otherwise, it holds a zero. $\odot$ represents the Hadamard product.

2. Network Structure

Inference Sub-Network and Statistics Sub-Network: In this section, we describe the additional information of our inference sub-network $\mathcal{L}$, statistics sub-network $\mathcal{Q}$, policy/generator π and discriminator $\mathcal{D}$ from Fig.2 to Fig.5. Table 1 specifies the details of temporal convolutional sub-module and fully connected layer. The fully convolutional sub-modules and deconvolutional sub-module in $\mathcal{L}$ and $\mathcal{Q}$ are pre-trained on ImageNet [1].

3. Additional Qualitative Results

We provide here additional qualitative examples on the SAP and ETH datasets, as in main paper, from Fig.6 to Fig.7. To better understand the multimodality of future paths that the proposed IDL captures, we highlight several individual examples from the prediction that obtains best ADE and two random selected predictions (random 1 and random 2). We also visualize the examples from deterministic IDL-NL2 and ground truth (G.T.). It is evidently that our IDL generates a diverse set of upcoming trajectories. For instance, the examples of best ADE in Fig.6 obtain results closest to the ground truth, while random 1 and random 2 cover other valid possibilities. These outcomes are attributed to explore the latent decision. In contrast, the IDL-NL2 baseline can only produce deterministic future paths, which disagree with ground truth considerably.

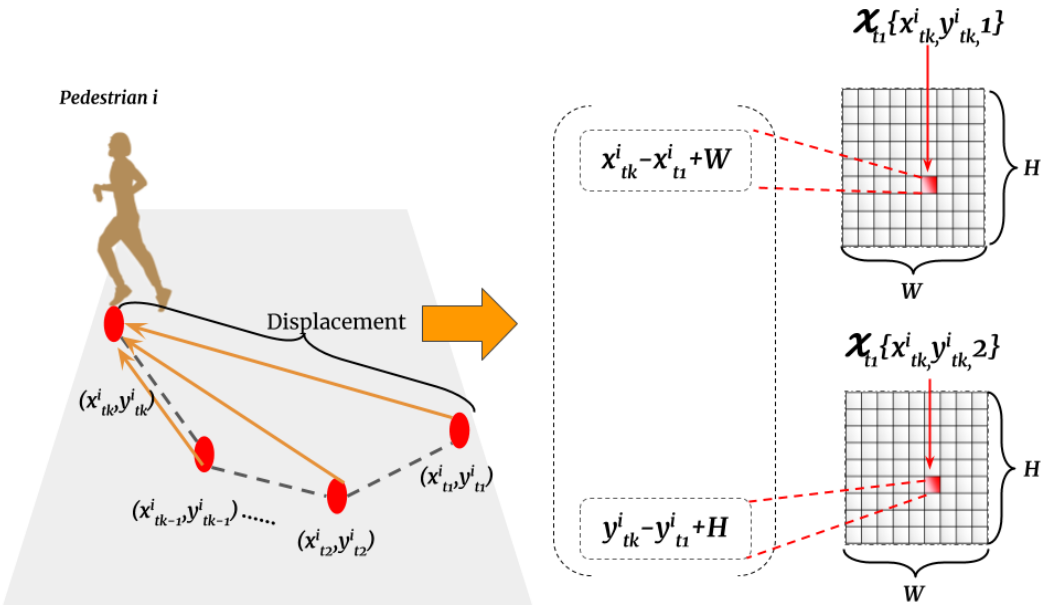

Figure 1. The example of constructing motion features.

	SAP		ETH & UCY	
	Kernel size	Output size	Kernel size	Output size
Temporal convolutional layer 1 in $\mathcal{L}$	Kernel length = 15 Number of kernels = 2048	$2048 \times 1 \times 4$	Kernel length = 2 Number of kernels = 2048	$2048 \times 1 \times 4$
Temporal convolutional layer 2 in $\mathcal{L}$	Kernel length = 2 Number of kernels = 1024	$1024 \times 1 \times 2$	Kernel length = 2 Number of kernels = 1024	$1024 \times 1 \times 2$
Temporal convolutional layer 1 in $\mathcal{Q}$	Kernel length = 15 Number of kernels = 256	$256 \times 1 \times 8$	Kernel length = 4 Number of kernels = 256	$256 \times 1 \times 3$
Temporal convolutional layer 2 in $\mathcal{Q}$	Kernel length = 8 Number of kernels = 64	$64 \times 1 \times 1$	Kernel length = 3 Number of kernels = 64	$64 \times 1 \times 1$
Fully Connect (FC) layer in $\mathcal{Q}$	–	1	–	1

Table 1. The configurations of temporal convolutional sub-module in both inference sub-network $\mathcal{L}$ and statistics sub-network $\mathcal{Q}$.

References

[1] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1

[2] Shuai Yi, Hongsheng Li, and Xiaogang Wang. Pedestrian Behavior Understanding and Prediction with Deep Neural Networks. In *European Conference on Computer Vision (ECCV)*, pages 263–279. Springer, 2016. 1

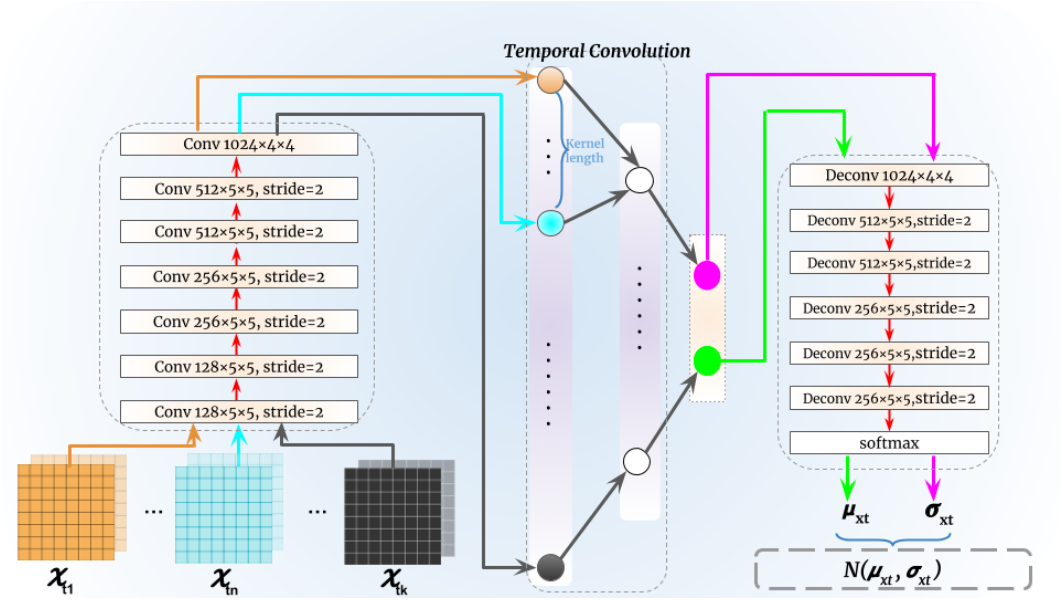

Figure 2. The network structure of the proposed inference sub-network $\mathcal{L}$. We first feed the motion features $\mathcal{X}_t (t \in \{t_1, t_2, ...t_k\})$ into a pre-trained fully convolutional sub-module one by one. The outputs will be taken as input of the temporal convolutional sub-module, which derives a tuple with a dimension of $1024 \times 1 \times 2$ in our experiments. We pass each unit with a dimension of $1024 \times 1 \times 1$ to a pre-trained deconvolutional module to produce the mean $\mu_{\mathcal{X}_t}$ and the variance $\sigma_{\mathcal{X}_t}$, respectively, for a Gaussian from which the latent decision samples. Notably we omit the padded zero and ReLU activation after each convolutional/deconvolutional layer of all sub-modules in the figure.

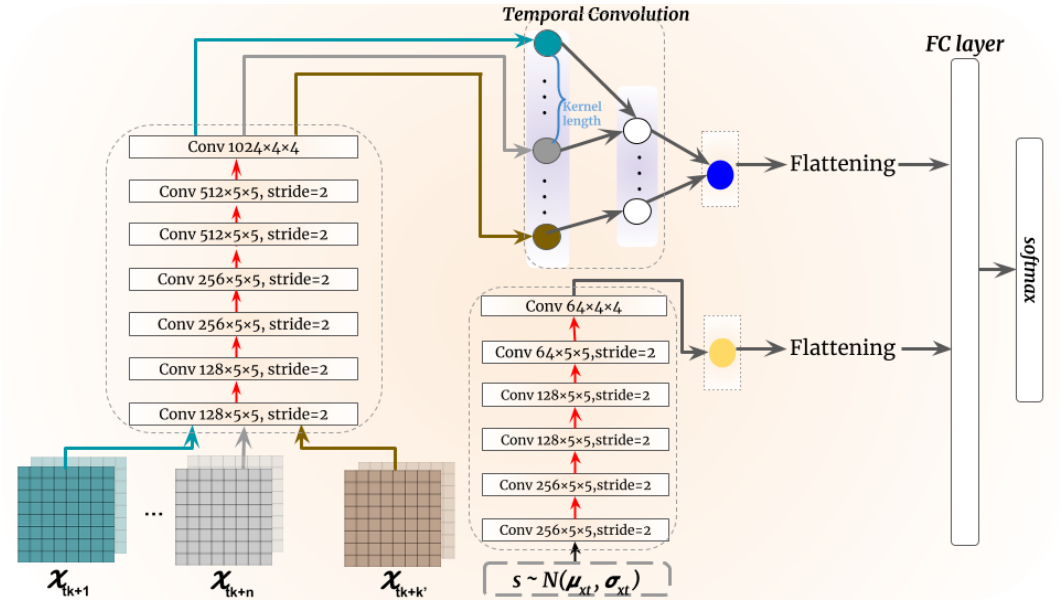

Figure 3. The pipeline of the proposed statistics sub-network $\mathcal{Q}$. The predicted motion features $\mathcal{X}_{t'} (t' \in \{t_{k+1}, t_{k+2}, ...t_{k+k'}\})$ are input to a pre-trained fully convolutional sub-module in a sequential manner. The output are read by a temporal convolutional module to produce a vector. We flatten the outcome from a dimension of $64 \times 1 \times 1$ to 64 vectors and then pass them to a fully connected (FC) layer with other 64 flattened vectors, which are from a different pre-trained convolutional sub-module that reads the sampled latent decision s. We insert a ReLU activation after each convolutional layer in each convolutional sub-module. We use zero-padding for all the convolutional operations.

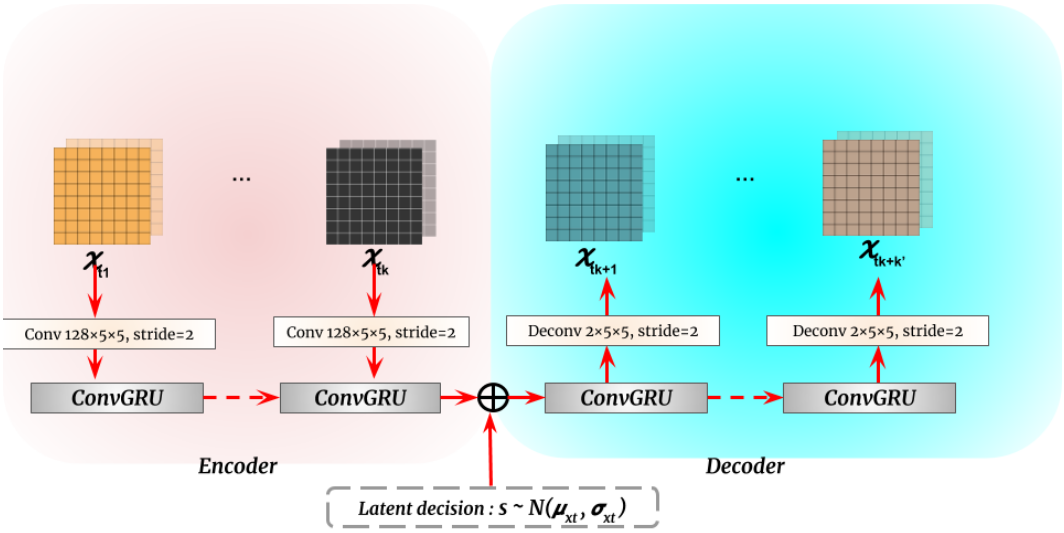

Figure 4. The structure of our policy/generator π. The zero-padding is used for each convolutiona/deconvolutonal layer. A ReLU activation is inserted after each convolutional/deconvolutional layer. We apply $256 \times 3 \times 3$ kernels with zero padding and a stride of 1 to the ConvGRU layer. A leaky ReLU activation with a negative slope of 0.1 replaces the tanh activation in ConvGRU.

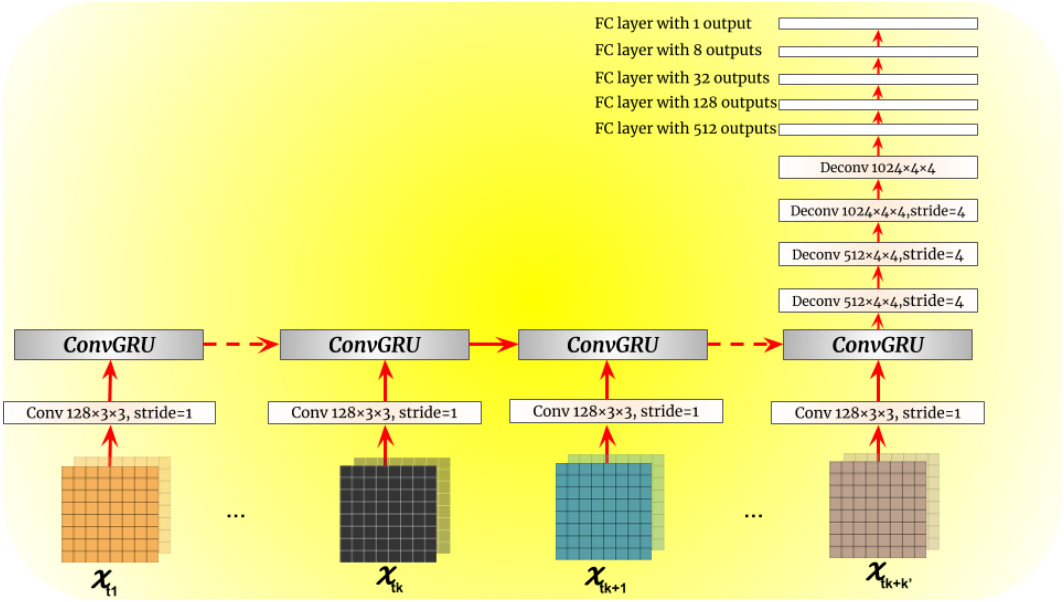

Figure 5. The pipeline of the proposed discriminator $\mathcal{D}$. We consider the last hidden states that are output of the ConvGRU layer by perceiving $\mathcal{X}_{t_{k+k'}}$ (or $\mathcal{GT}_{t_{k+k'}}$) as the descriptor of $[\mathcal{X}_t, \mathcal{X}_{t'}]$ (or $[\mathcal{X}_t, \mathcal{GT}_{t'}]$). During our experiment, the ConvGRU uses $256 \times 3 \times 3$ kernels with a stride of 1 and zero padding. We use a leaky ReLU activation with a negative slope of 0.1 instead of tanh activation of the ConvGRU. A ReLU activation is employed after each deconvolutional layer and fc layer. All the convolutional and deconvolutional operations use zero-padding.

Figure 6. The qualitative comparisons on SAP dataset.

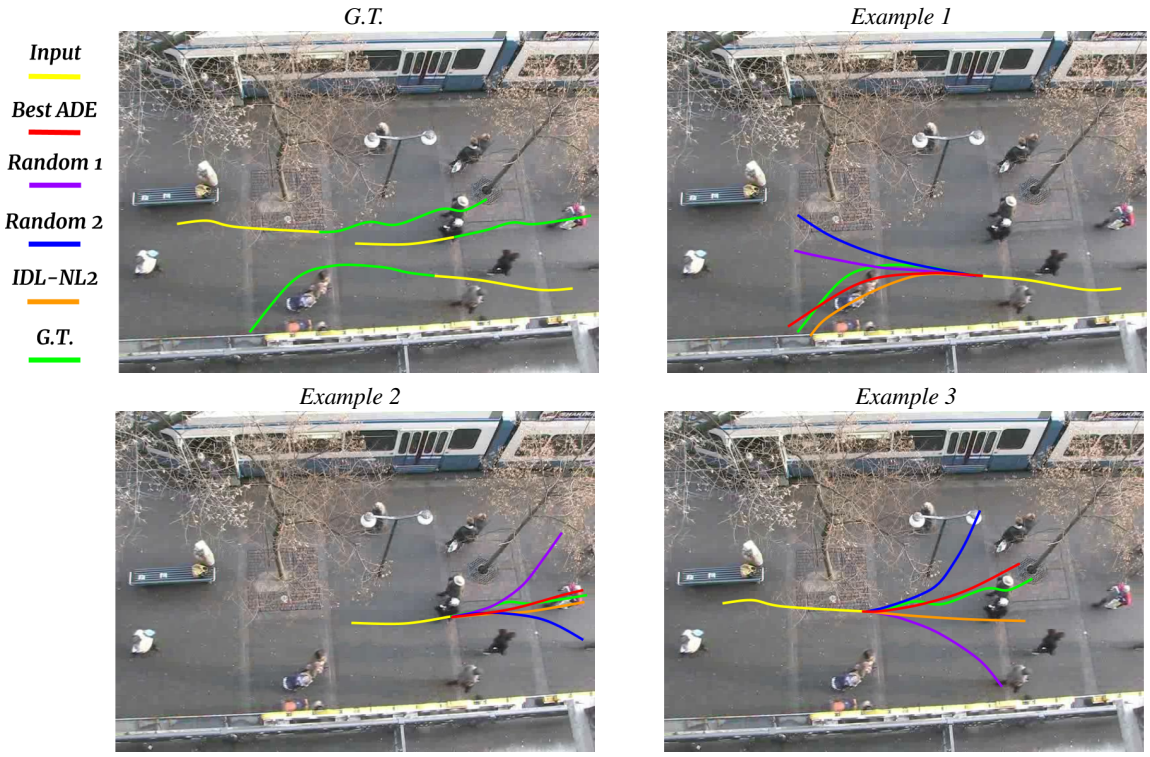

Figure 7. The visual results on ETH dataset.