Trajectron and Trajectron++ Learning Poster

Tags: Robot Autonomy, Multi-Agent Learning, Trajectory Prediction

(Reference:
1. Ivanovic, Boris, and Marco Pavone. "The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.
2. Salzmann, Tim, et al. "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data." Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16. Springer International Publishing, 2020.)

Trajectron++

Trajectron

1. Graph-structured Model
2. For Future Trajectories Prediction of All Agents (Purpose)
3. Structure $\begin{cases}\text{Encoder: Recurrent Model + Additive Attention}\\ \text{Decoder: CVAE Framework + Recurrent Model}\end{cases}$
4. Advantages
   a. Dynamic: Time-varying agents' number.
   b. Multimodal: Output possible distinct future trajectories.
   c. Heterogeneous Data: Environmental info like obstacles.
   d. Incorporating future Ego-Agent motion plans.

Generating Distributions of Trajectories:

1. CVAE: $p(\mathbf{y} \mid \mathbf{x}) = \sum p_\psi(\mathbf{y} \mid \mathbf{x}, z) p_\theta(z \mid \mathbf{x})\, dz$
2. Thus: $h^t_{i,enc} = [h^t_{i,edges}; h^t_{i,node}]$

$\phi = MLP\left([h^t_{i,enc}; h^{t+}_{i,node}]; W_{\phi,C_i}\right)$ $\theta = MLP(h^t_{i,enc}; W_{\theta,C_i})$

Encourage similar observations with similar latent representations.

$$z \sim \begin{cases} q_\phi(z \mid \mathbf{x}_i, \mathbf{y}_i), & \text{for training} \\ p_\theta(z \mid \mathbf{x}_i), & \text{for testing} \end{cases}$$

$$\hat{\mathbf{y}}^t_i \sim GMM\left(LSTM\left([\hat{\mathbf{y}}^{t-1}_i, z, h^t_{i,enc}]; W_{\psi,C_i}\right)\right)$$

Training:

$$\max_{\phi,\theta,\psi} \sum_{i=1}^{N} \mathbb{E}_{z \sim q_\phi(z \mid \mathbf{x}_i, \mathbf{y}_i)}\left[\log p_\psi(\mathbf{y}_i \mid \mathbf{x}_i, z)\right] - \beta D_{KL}\left(q_\phi(z \mid \mathbf{x}_i, \mathbf{y}_i) \,\|\, p_\theta(z \mid \mathbf{x}_i)\right)$$

KL divergence: encourage predictable.

Edge Encoding:

1. LSTM encoder for edges of same type:

$$e^t_{i,k} = \left[\mathbf{x}^t_i; \sum_{j \in N_k(i)} \mathbf{x}^t_j\right]$$

$$h^t_{i,k} = LSTM\left(h^{t-1}_{i,k}, e^t_{i,k}; W_{EE,k}\right)$$

For a certain LSTM encoder for certain edge type, a fixed architecture is used to handle a variable number of neighboring nodes while preserving count information.

2. Additive Attention for Edge Encodings Aggregation:

a. Modulation: $M = \min\{A * E + R * E, 1\}$

$$\widetilde{h^t_{i,k}} = h^t_{i,k} \odot \min\left\{\sum_{j \in N_k(i)} M[t,i,j], 1\right\}$$

b. Additive Attention:

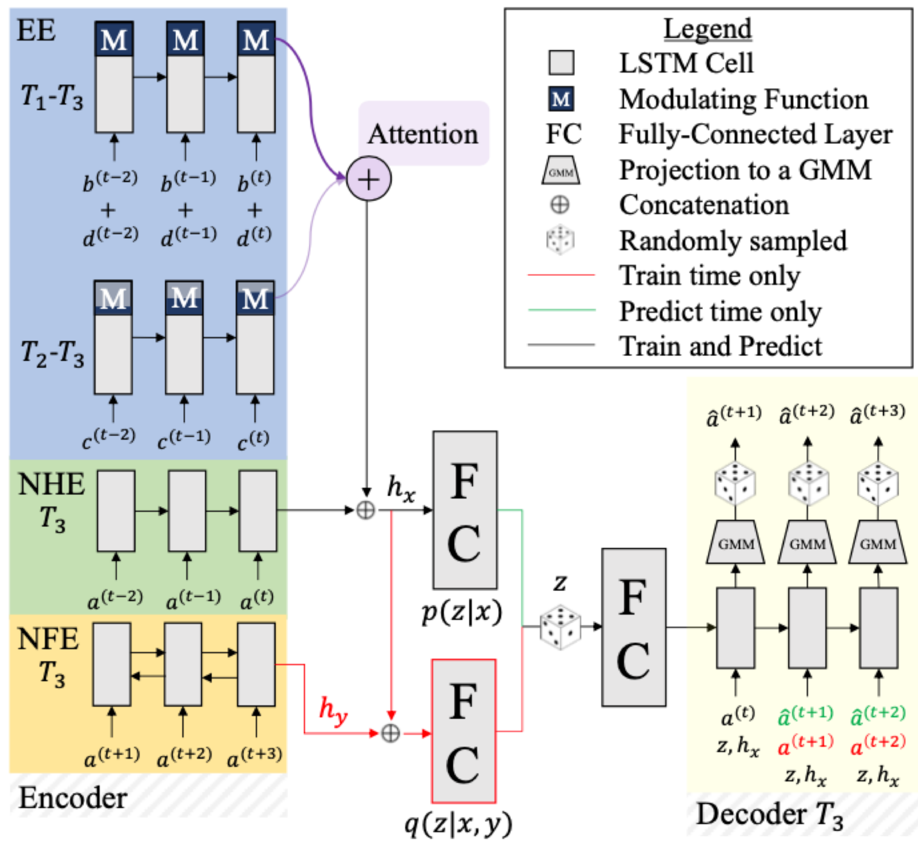
$$s^t_{ik} = v^T_{C_i} \tanh\left(W_{1,C_i} \widetilde{h^t_{i,k}} + W_{2,C_i} h^t_{i,node}\right)$$

$$a^t_i = \text{softmax}([s^t_{i1}, \ldots, s^t_{iK}]) \in \mathbb{R}^K$$

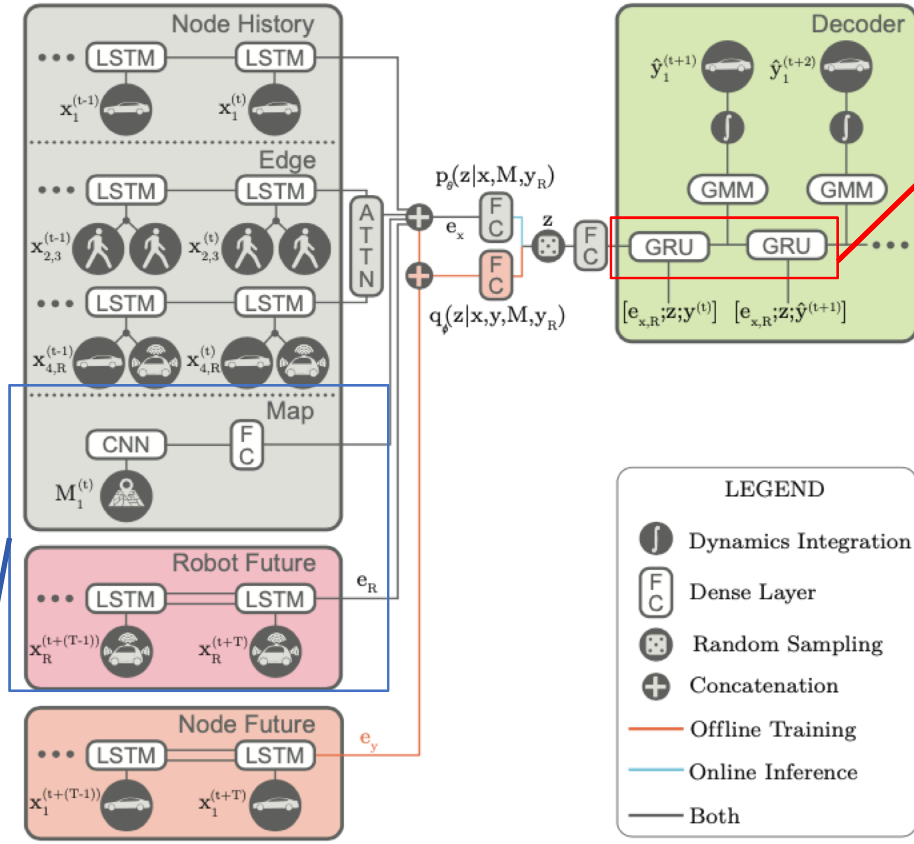
$$h^t_{i,edges} = \sum_{k=1}^{K} a^t_{ik} \odot \widetilde{h^t_{i,k}}$$

The node history encoding attends to modulated edge encodings to look for a combination of edges that is most relevant to an agent's current state.

Trajectron



Trajectron++



Change Point 2: LSTMs of decoder change to simpler GRUs for <u>more efficient computation</u>.

Change Point 3: Training

$$\max_{\phi,\theta,\psi} \sum_{i=1}^{N} \mathbb{E}_{z\sim q_\phi(\cdot|\mathbf{x}_i,\mathbf{y}_i)}\left[\log p_\psi(\mathbf{y}_i \mid \mathbf{x}_i, z)\right]$$
$$- \beta D_{KL}\left(q_\phi(z \mid \mathbf{x}_i,\mathbf{y}_i) \,\|\, p_\theta(z \mid \mathbf{x}_i)\right) + \boxed{\alpha I_q(\mathbf{x}; z),}$$

Mutual information between observations and latent representations.

Change Point 1: Heterogeneous data and future Ego-Agent motion plans are taken into account.

a. A CNN followed by a fully connected layer is used for heterogeneous data encoding.

b. Incorporating future Ego-Agent motion plans allows for the <u>evaluation of a set of motion primitives</u> with respect to possible responses from other agents.

Summarization and Personal Thinking

Summarization:

1. Innovation:
   a. Graphical Scene Representation
   b. CVAE Framework for Trajectories Prediction:
      b.1 Introduce discrete latent variable z, and z being discrete aids in interpretability.
      b.2 Predictable: KL divergence in training loss encourages predictable.
   c. Dynamic and Variable: Deal with situation of time-varying agents' number.
   d. Multimodal: Generate many possible distinct future trajectories.
   e. Heterogenous data encoding for environmental information modeling. (Trajectron++)
   f. Consider about future Ego-Agent motion plans. (Dynamic Constraints, Trajectron++)
2. Limitation:
   Because it's agnostic to how a forecasting trajectory might affect downstream planning, inappropriate combination of trajectories prediction and downstream planner (e.g. weights of different trajectories) may lead to bad planning results.

Future Work and Personal thinking:

1. Harnessing properly trajectory prediction in downstream planning.
2. Building more general prediction models that can be transferred to different situations.
3. Making models robust to uncertainty:
   e.g. Uncertainty Calibration and Randomness Modeling
   e.g. Directly use detections and affinity (similarity between frames) as prediction inputs for reducing prediction problems from tracking error. (ASL's work on CVPR 22)
   e.g. Explicitly modeling uncertainty by building benchmark dataset with tracking error data (for example).