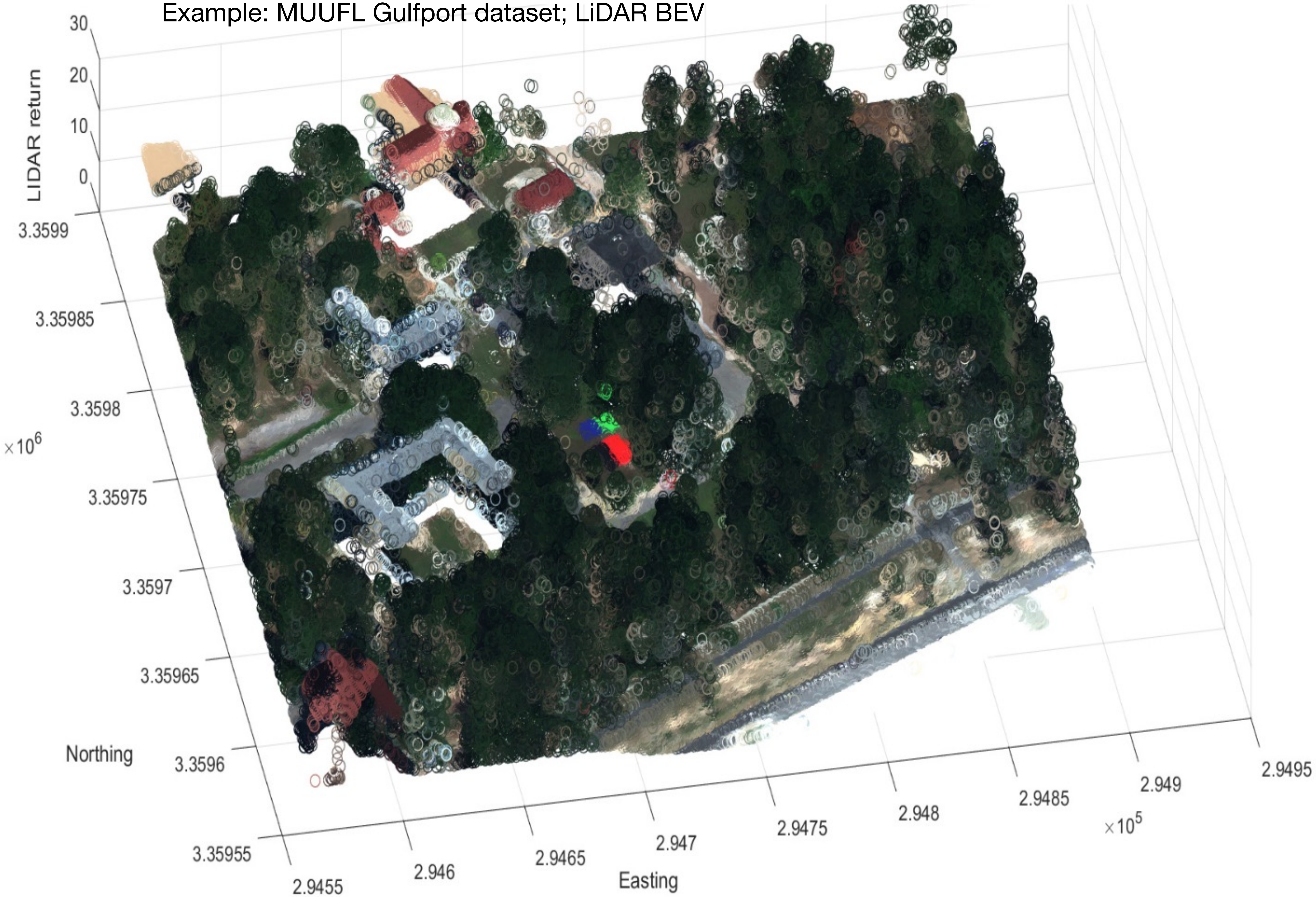
DeepRob

Lecture 20
Video Processing
University of Michigan | Department of Robotics

Recall: 3D Vision

3D Point clouds



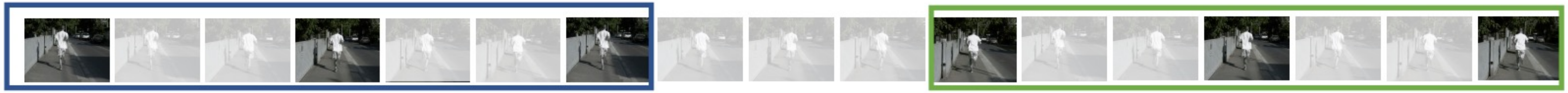
Example: MUUFL Gulfport dataset; LiDAR BEV

Videos – The temporal dimension

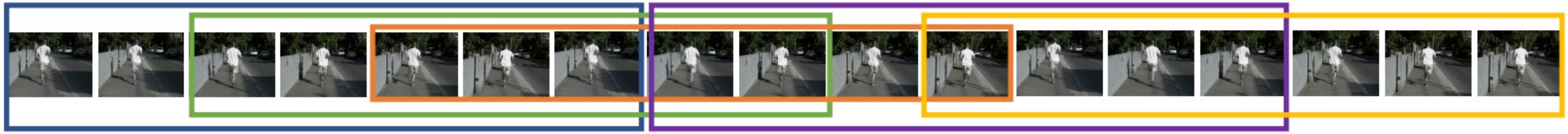
Video clips

Raw video: Long, high FPS

Training: Train model to classify short **clips** with low FPS



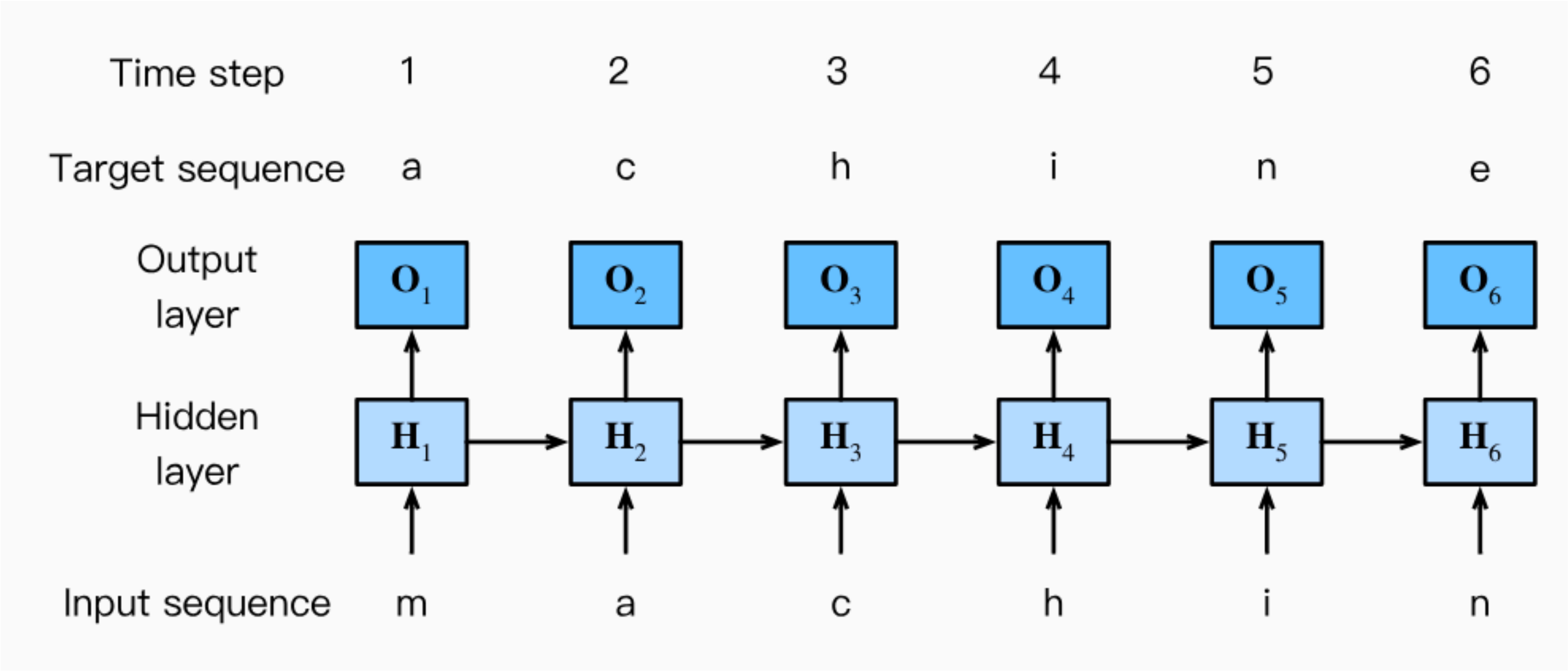
Testing: Run model on different clips, average predictions



Adapted from Prof. Johnson's slides

Videos – The temporal dimension

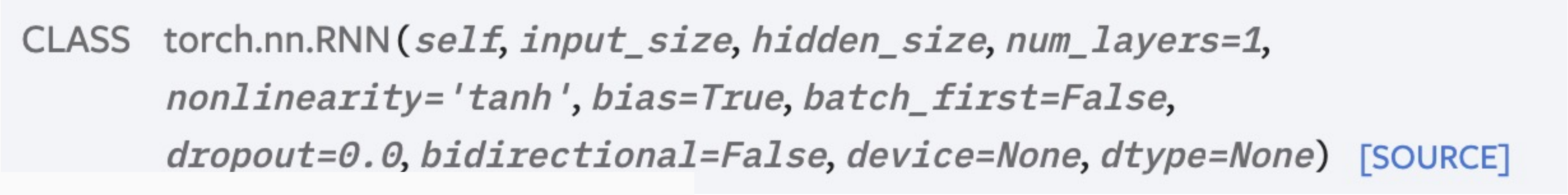
Sequence prediction, classification, translation, etc……



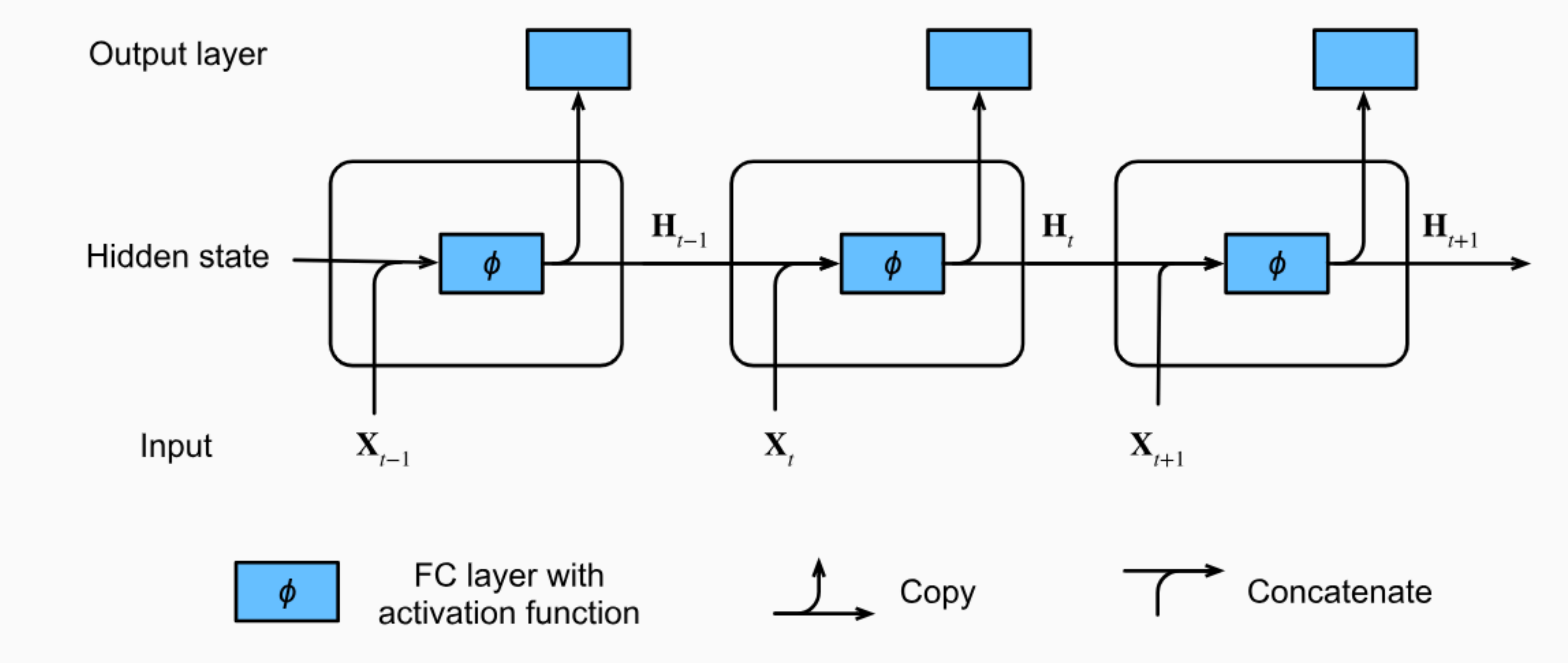
https://d2l.ai/chapter_recurrent-neural-networks/rnn.html#recurrent-neural-networks-with-hidden-states

RNN

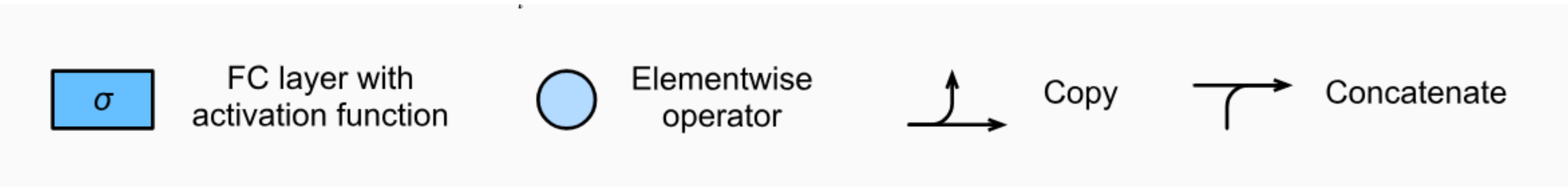
- Recurrent Neural Network



CLASS torch.nn.RNN(*self*, *input_size*, *hidden_size*, *num_layers=1*, *nonlinearity='tanh'*, *bias=True*, *batch_first=False*, *dropout=0.0*, *bidirectional=False*, *device=None*, *dtype=None*) [SOURCE]



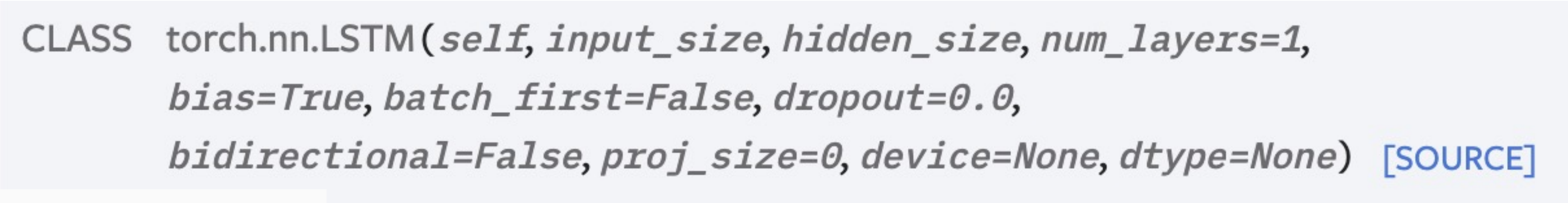
$$\mathbf{H}_t = \phi(\mathbf{X}_t\mathbf{W}_{\mathrm{xh}} + \mathbf{H}_{t-1}\mathbf{W}_{\mathrm{hh}} + \mathbf{b}_{\mathrm{h}})$$



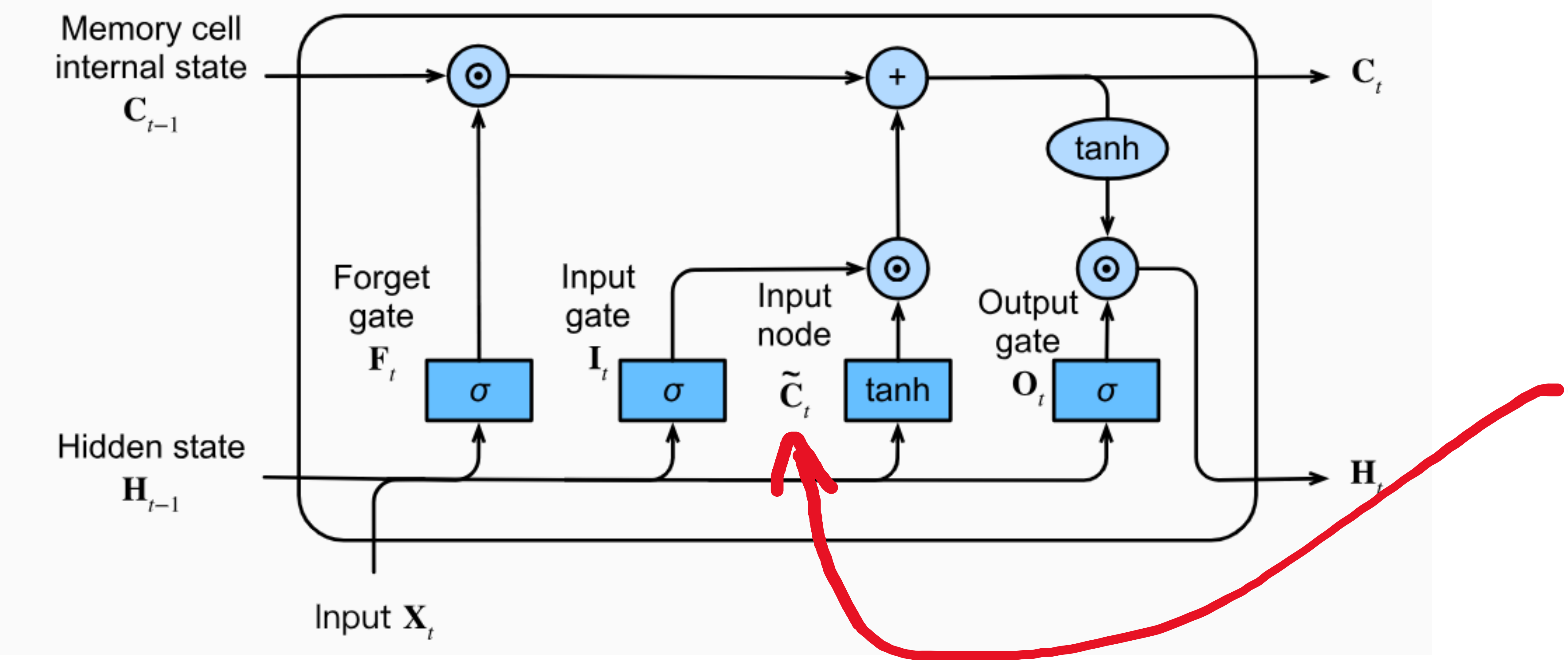
https://d2l.ai/chapter_recurrent-neural-networks/rnn.html#recurrent-neural-networks-with-hidden-states

LSTM

- Long Short Term Memory

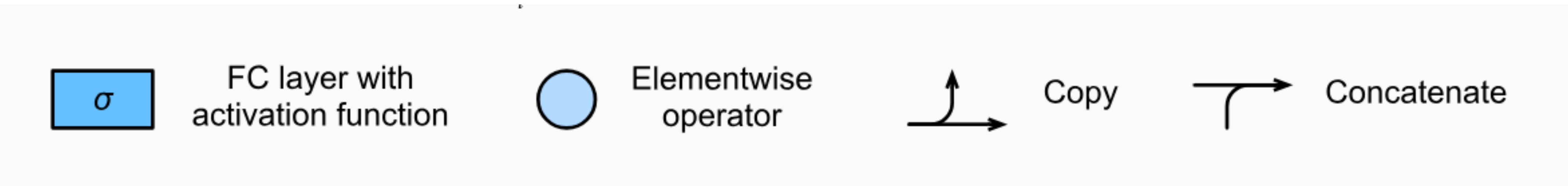


CLASS torch.nn.LSTM(*self*, *input_size*, *hidden_size*, *num_layers=1*, *bias=True*, *batch_first=False*, *dropout=0.0*, *bidirectional=False*, *proj_size=0*, *device=None*, *dtype=None*) [SOURCE]



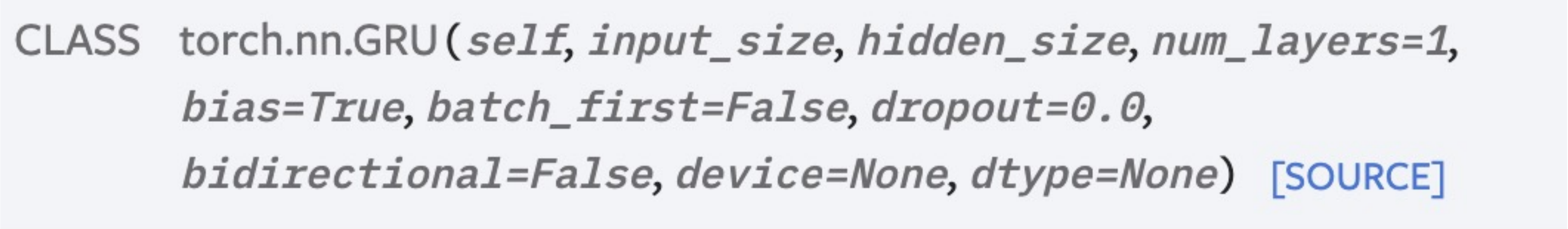
Memory cell internal state $\mathbf{C}_{t-1}$
$\mathbf{C}_t$
tanh
Forget gate $\mathbf{F}_t$
Input gate $\mathbf{I}_t$
Input node $\tilde{\mathbf{C}}_t$
Output gate $\mathbf{O}_t$
σ
σ
tanh
σ
Hidden state $\mathbf{H}_{t-1}$
$\mathbf{H}_t$
Input $\mathbf{X}_t$

$$
\begin{aligned}
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\
f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\
g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \\
o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

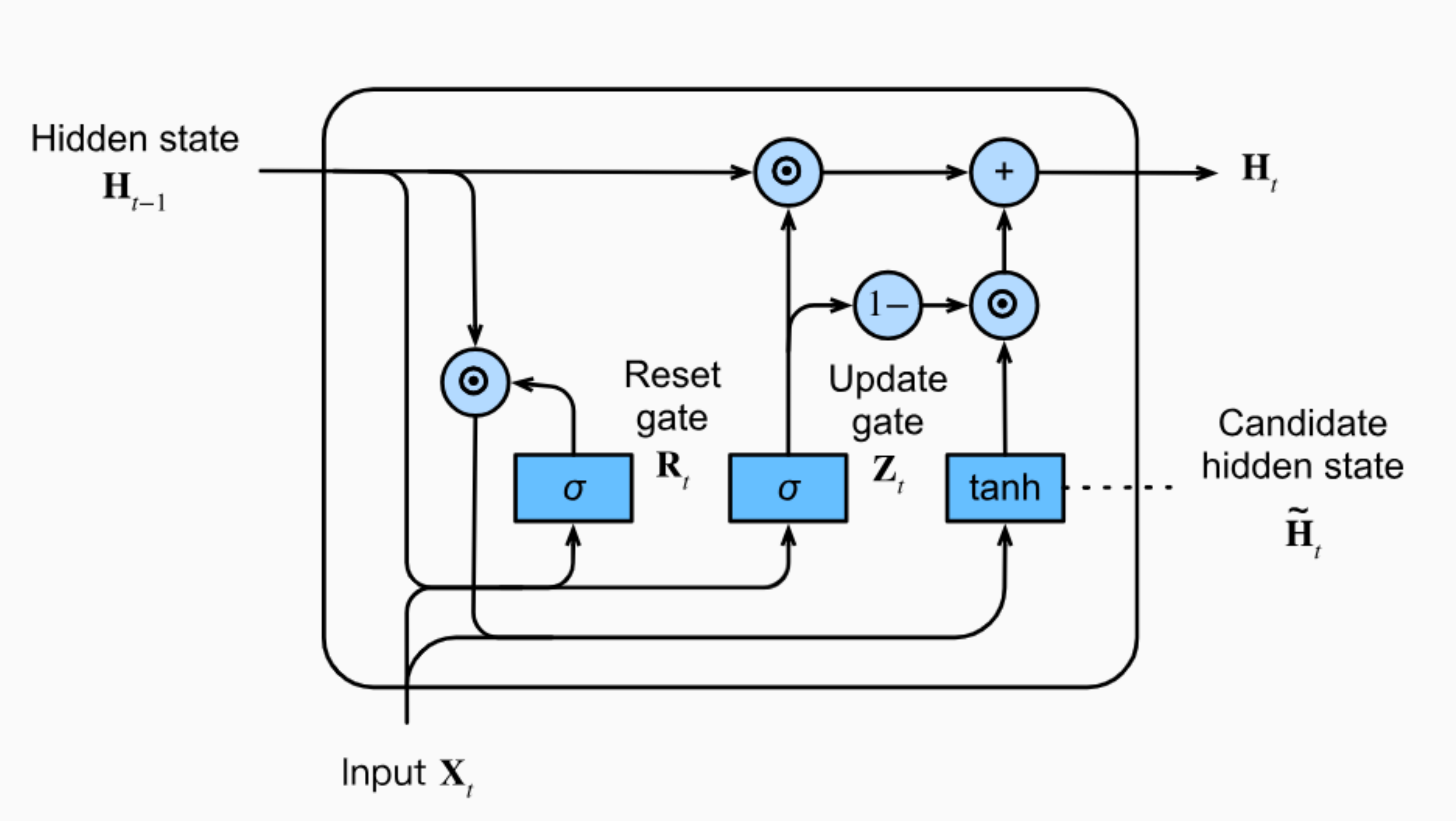


https://d2l.ai/chapter_recurrent-modern/lstm.html

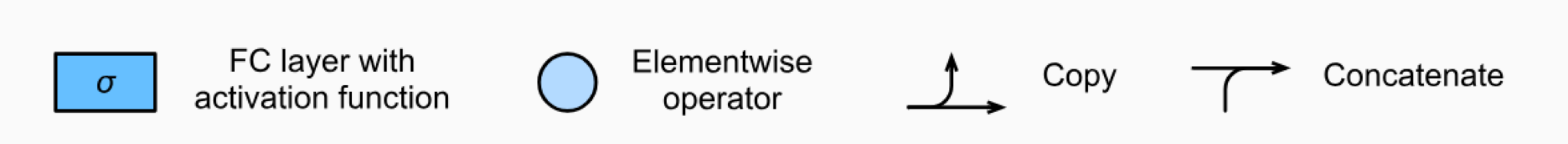
GRU (Gated Recurrent Unit)



CLASS torch.nn.GRU(*self*, *input_size*, *hidden_size*, *num_layers=1*, *bias=True*, *batch_first=False*, *dropout=0.0*, *bidirectional=False*, *device=None*, *dtype=None*) [SOURCE]

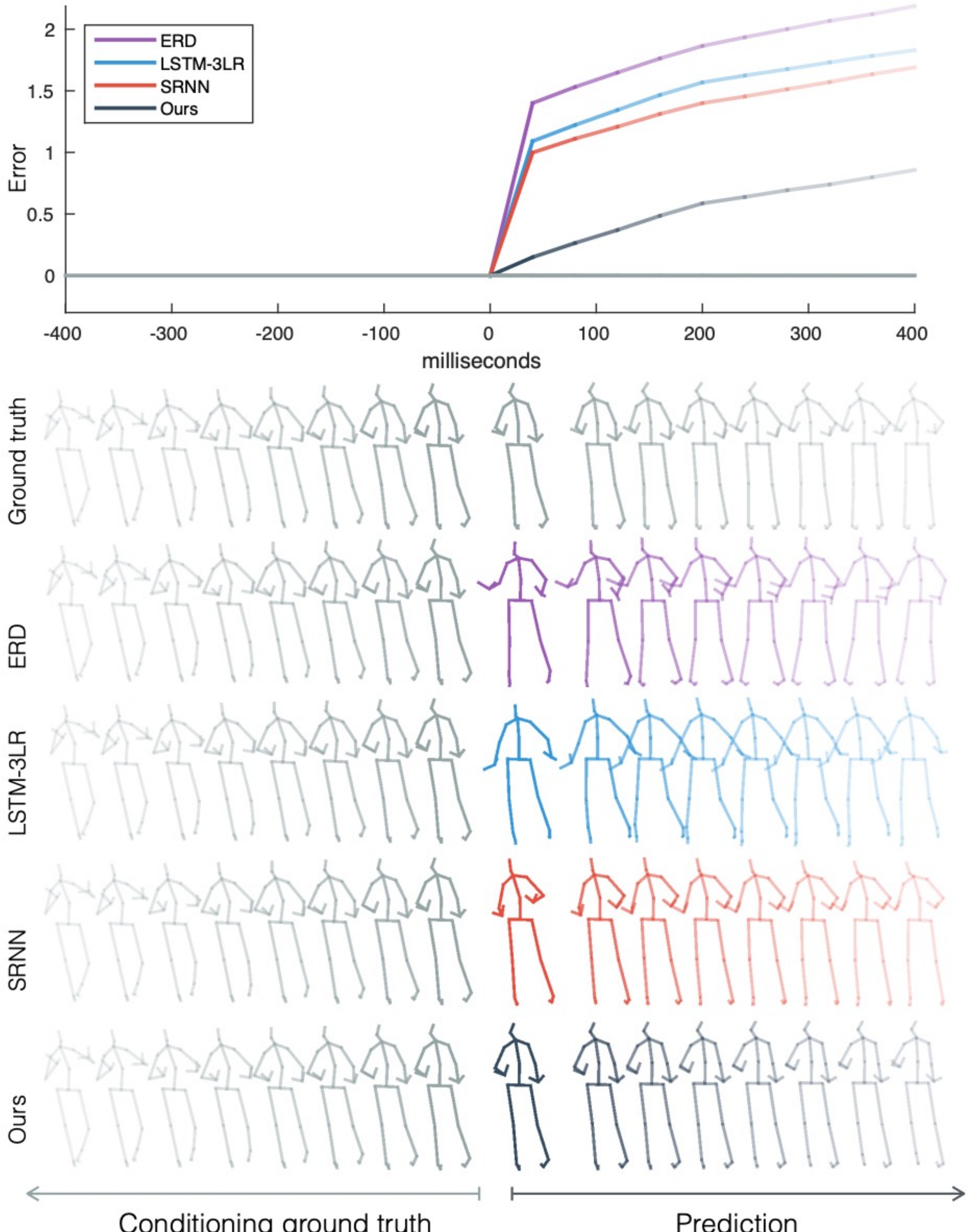


$$
\begin{aligned}
r_t &= \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{(t-1)} + b_{hr}) \\
z_t &= \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{(t-1)} + b_{hz}) \\
n_t &= \tanh(W_{in}x_t + b_{in} + r_t \odot (W_{hn}h_{(t-1)} + b_{hn})) \\
h_t &= (1 - z_t) \odot n_t + z_t \odot h_{(t-1)}
\end{aligned}
$$



https://d2l.ai/chapter_recurrent-modern/gru.html

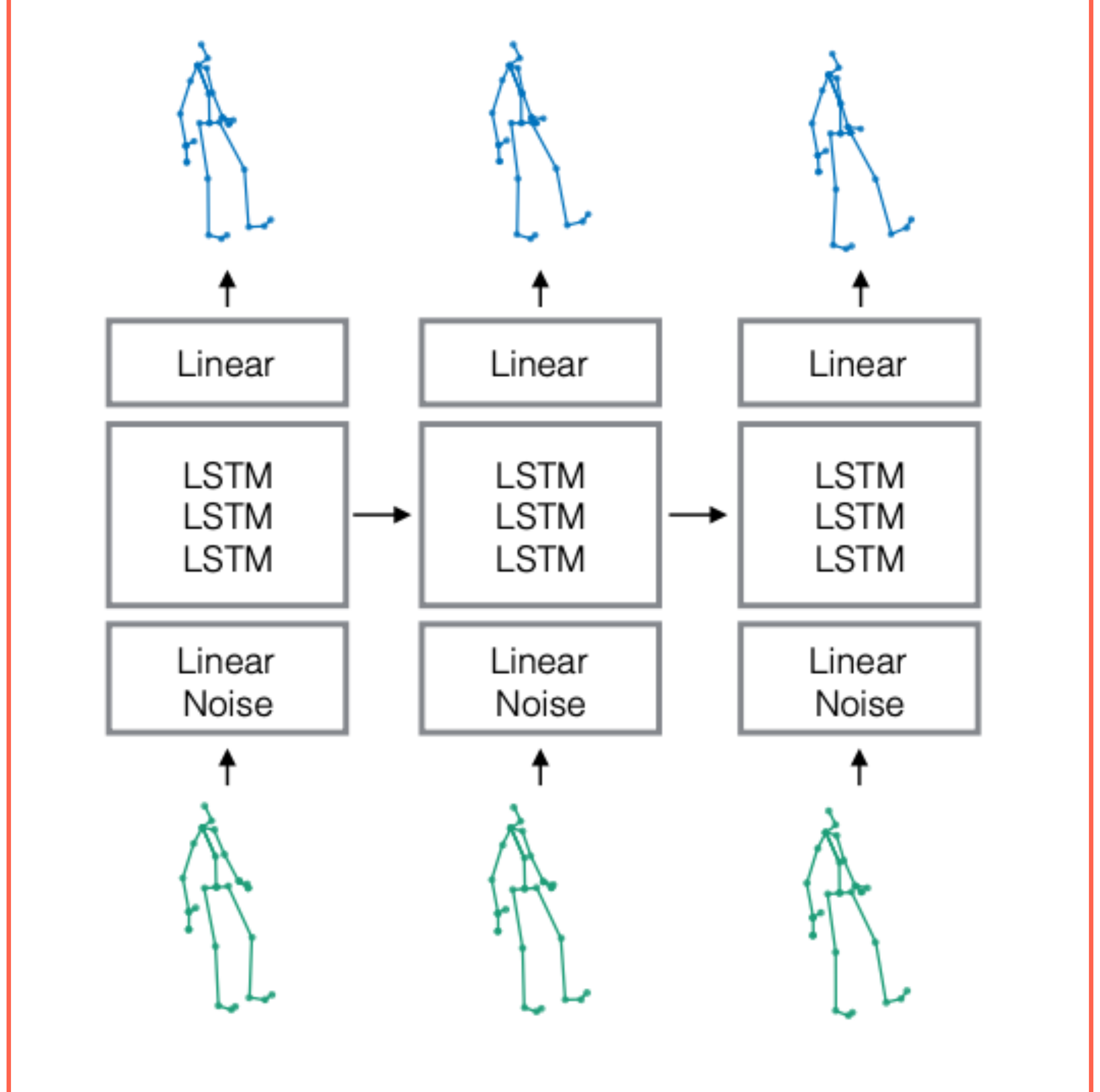
Long-term Human Motion Prediction



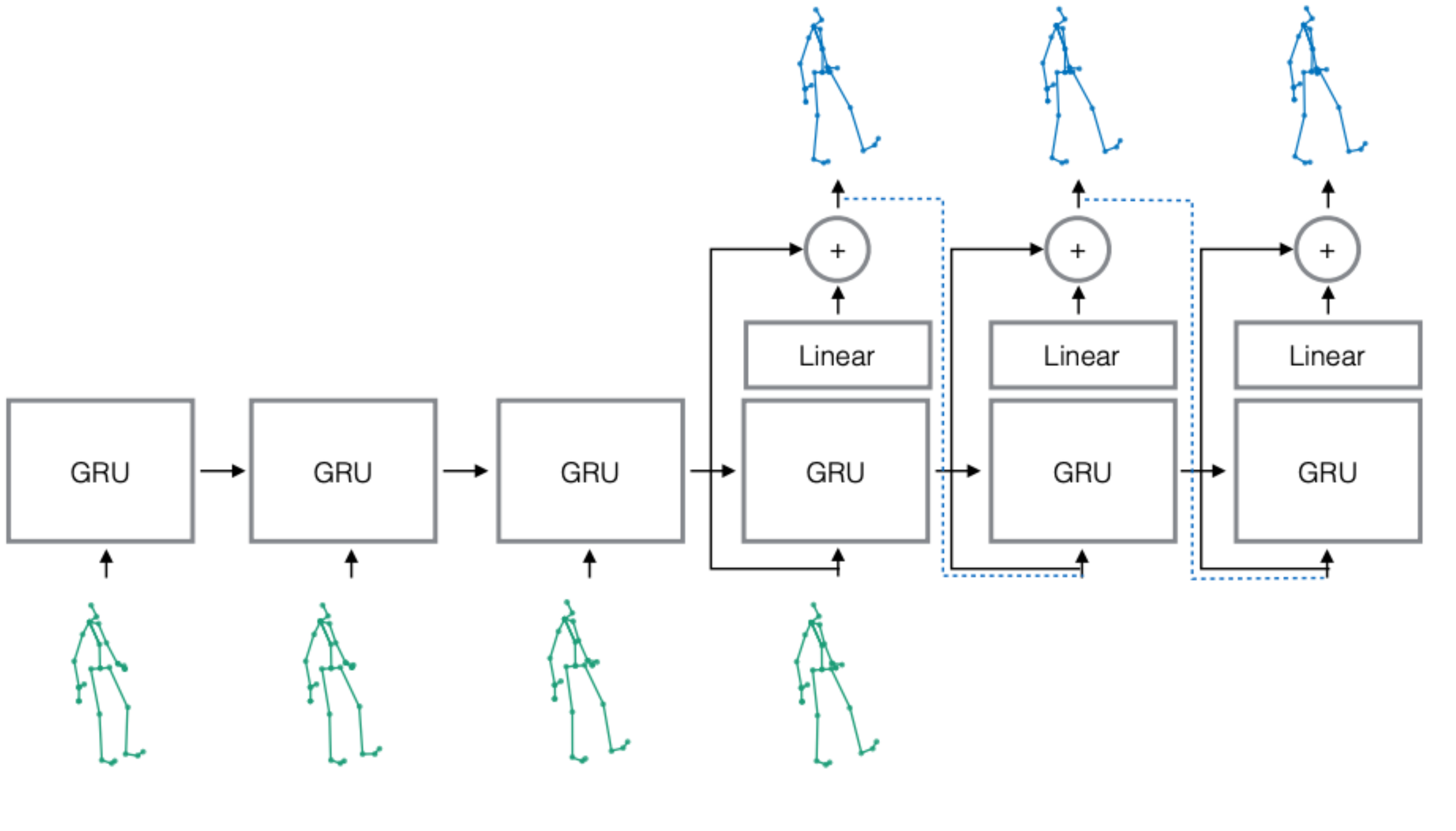
On human motion prediction using recurrent neural networks, cvpr 2017

Long-term Human Motion Prediction

LSTM-3LR

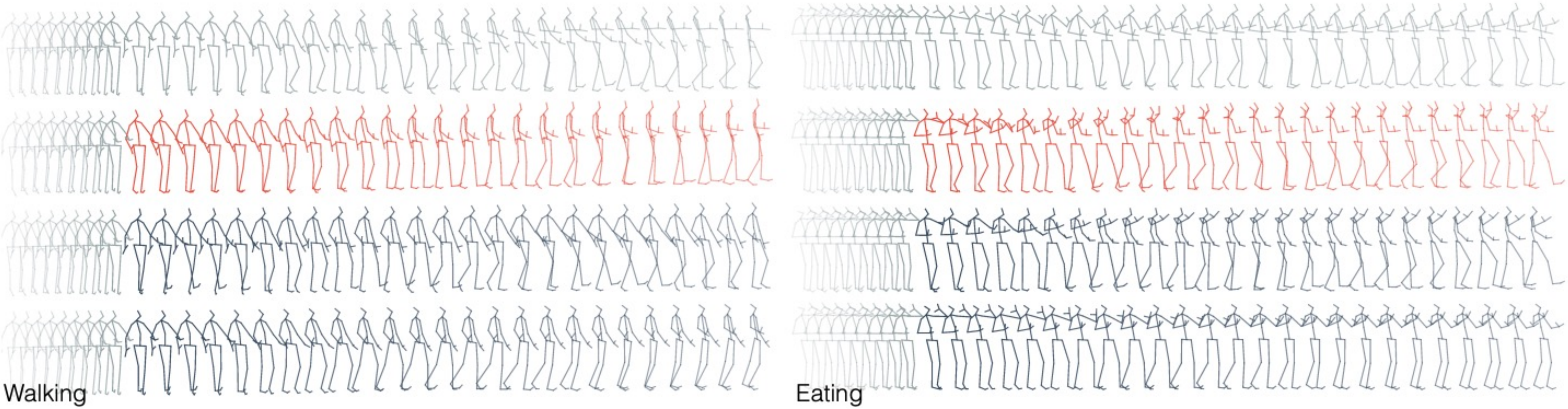


Seq2seq architecture



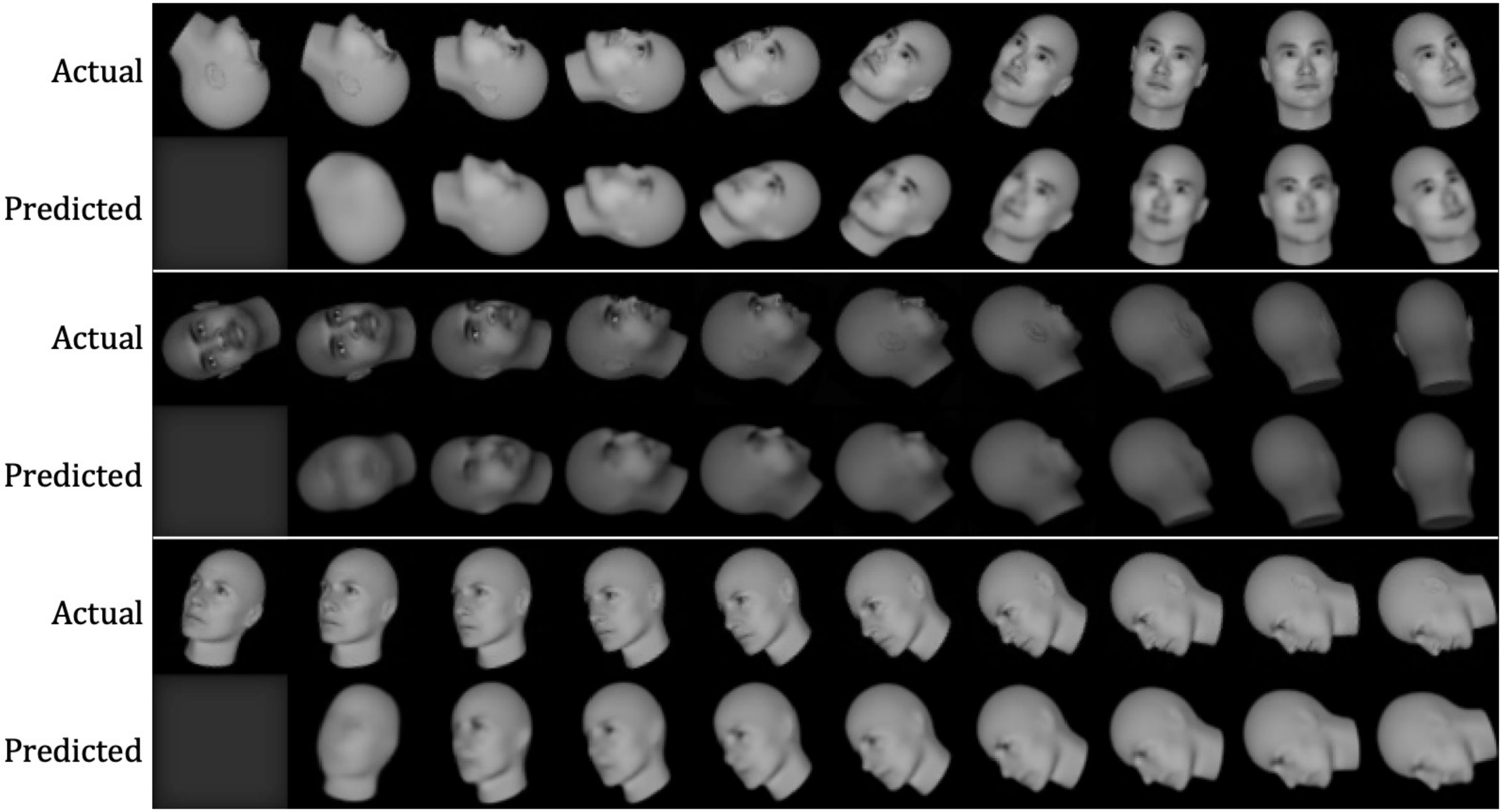
On human motion prediction using recurrent neural networks, cvpr 2017

Long-term Human Motion Prediction



PredNet

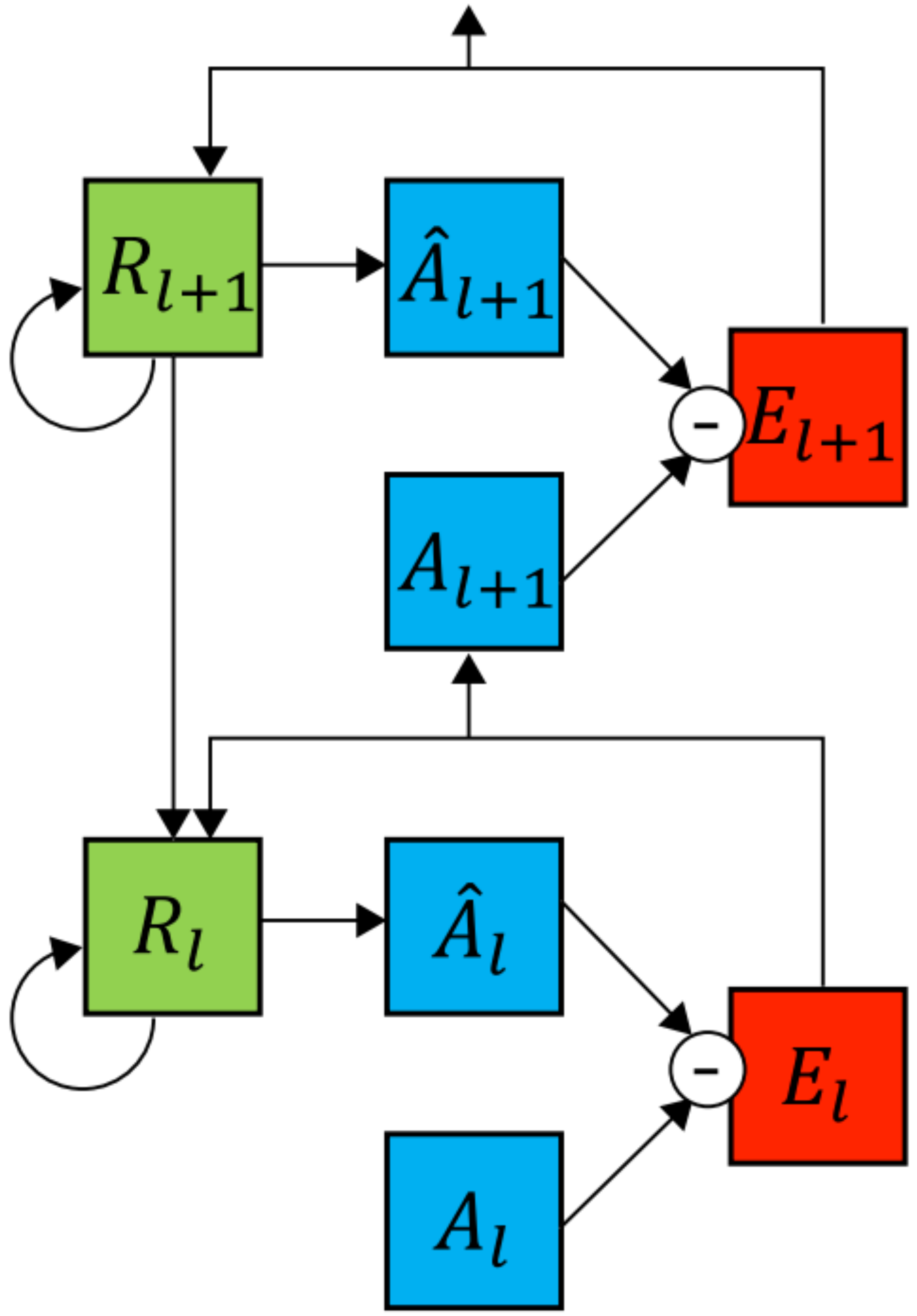
- Next-frame prediction

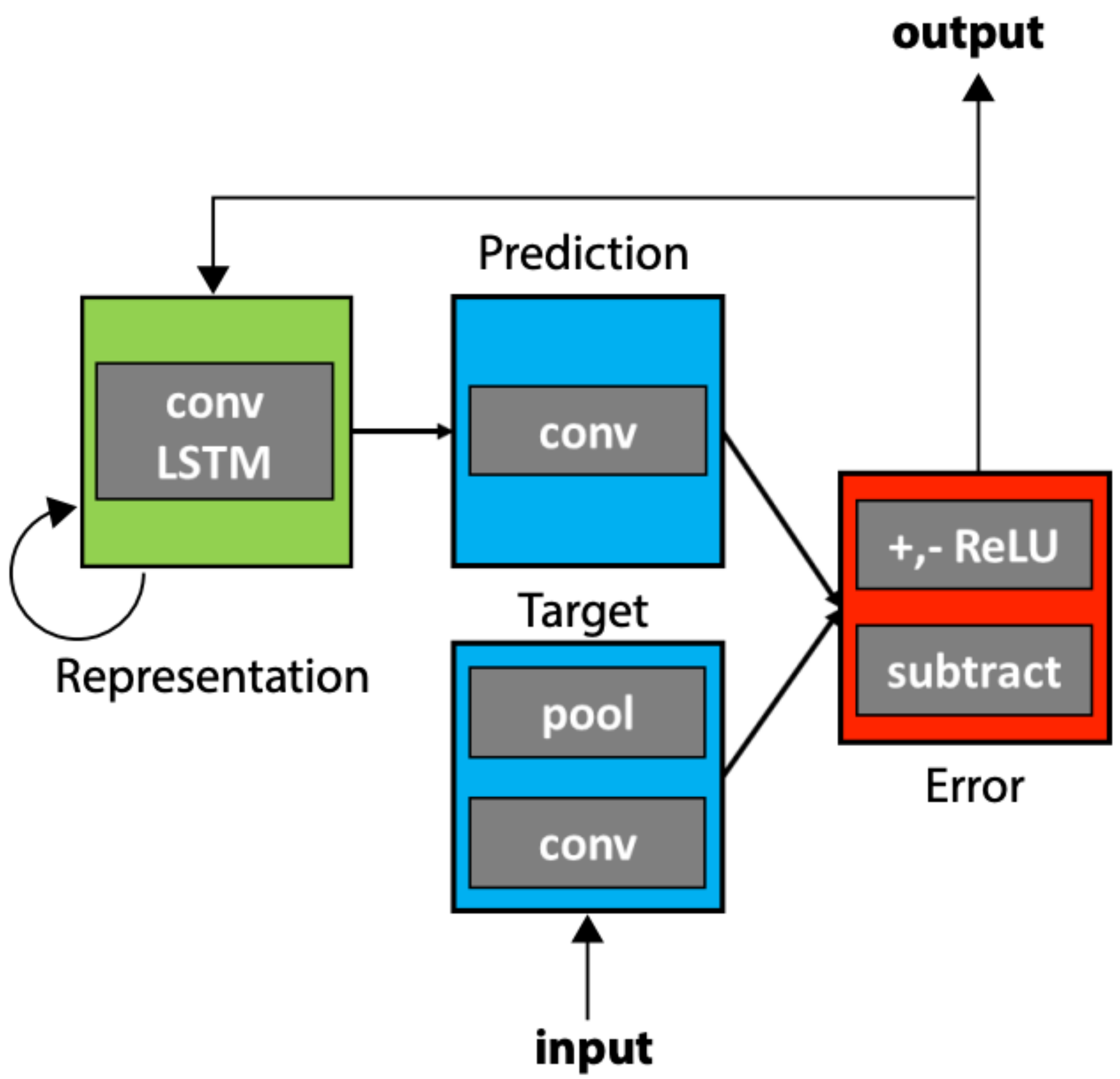


DEEP PREDICTIVE CODING NETWORKS FOR VIDEO PREDICTION AND UNSUPERVISED LEARNING, ICLR 2017

PredNet

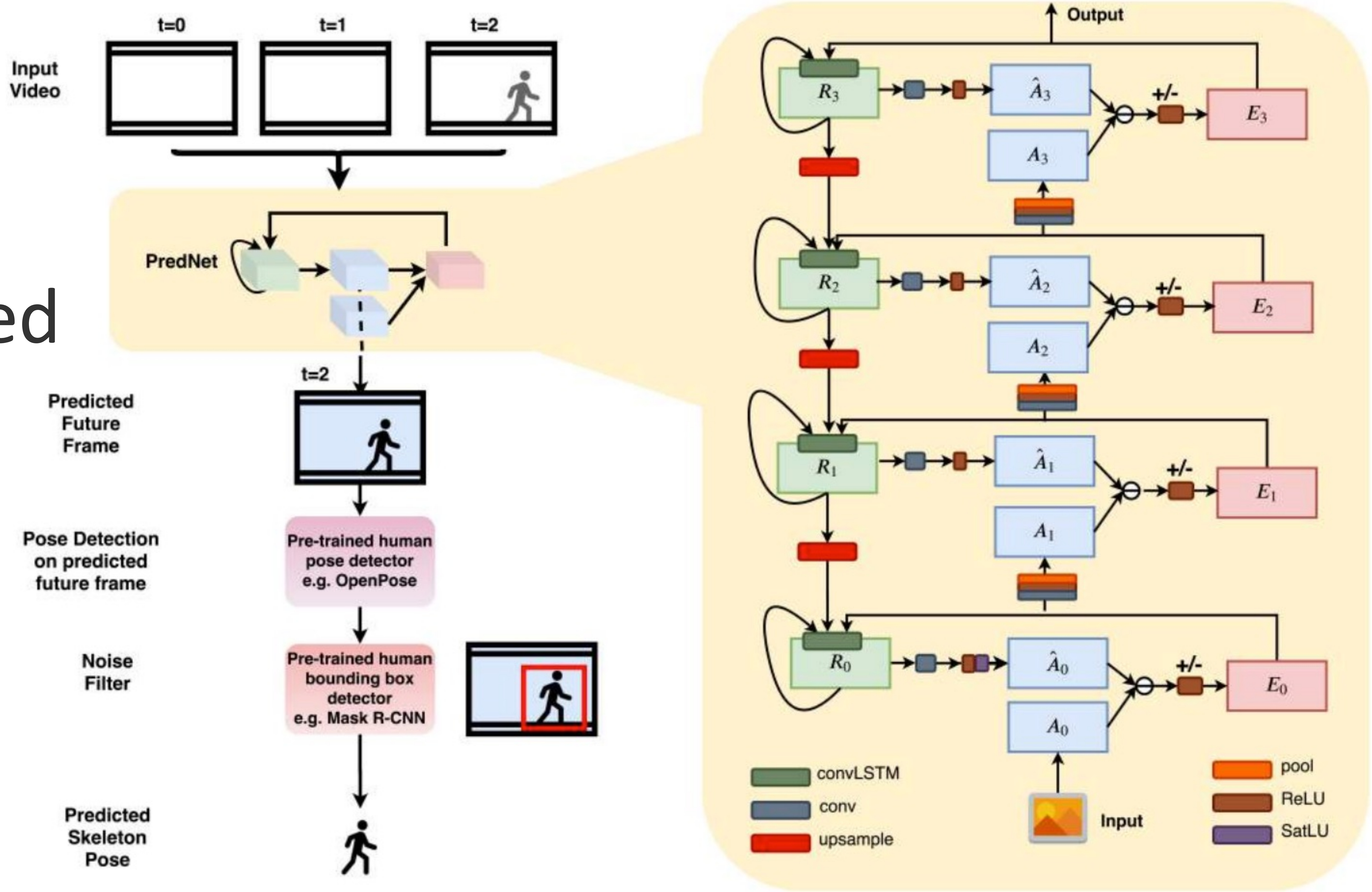
- Next-frame prediction





DEEP PREDICTIVE CODING NETWORKS FOR VIDEO PREDICTION AND UNSUPERVISED LEARNING, ICLR 2017

PredNet-based prediction application

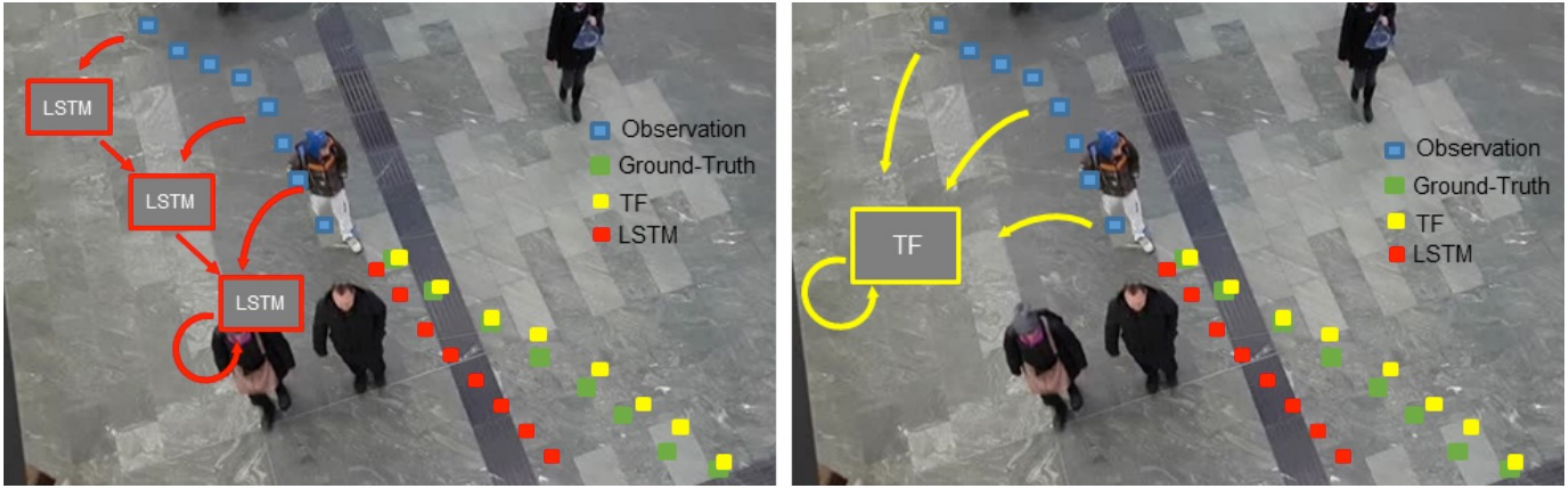


Unsupervised Pedestrian Pose Prediction A deep predictive coding network based approach for autonomous vehicle perception, RA Magazine, 2019

PredNet frame prediction

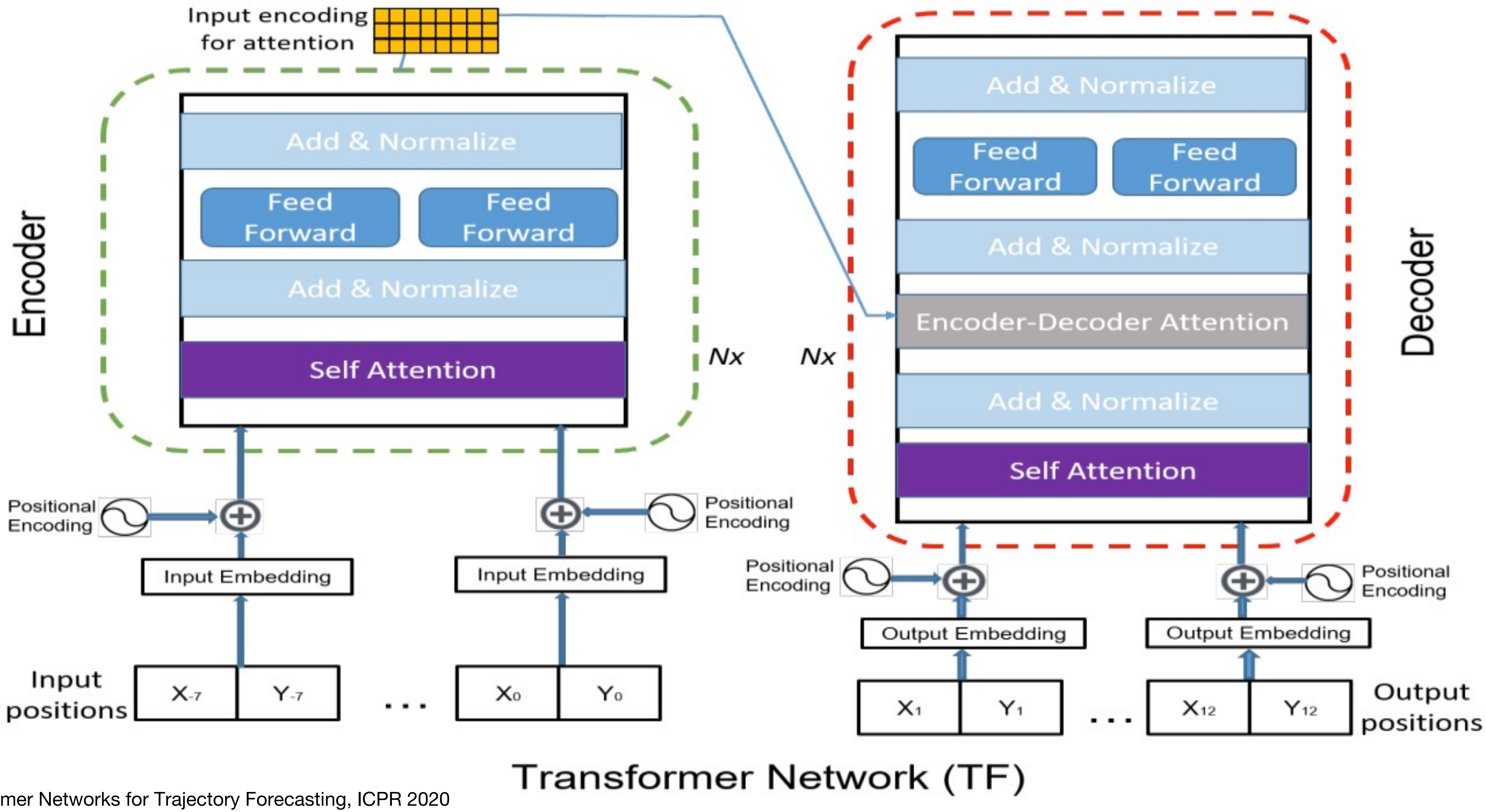
https://coxlab.github.io/prednet/

Transformers for trajectory prediction



Transformer Networks for Trajectory Forecasting, ICPR 2020

Transformers for trajectory prediction



Transformer Networks for Trajectory Forecasting, ICPR 2020

Collision-aware

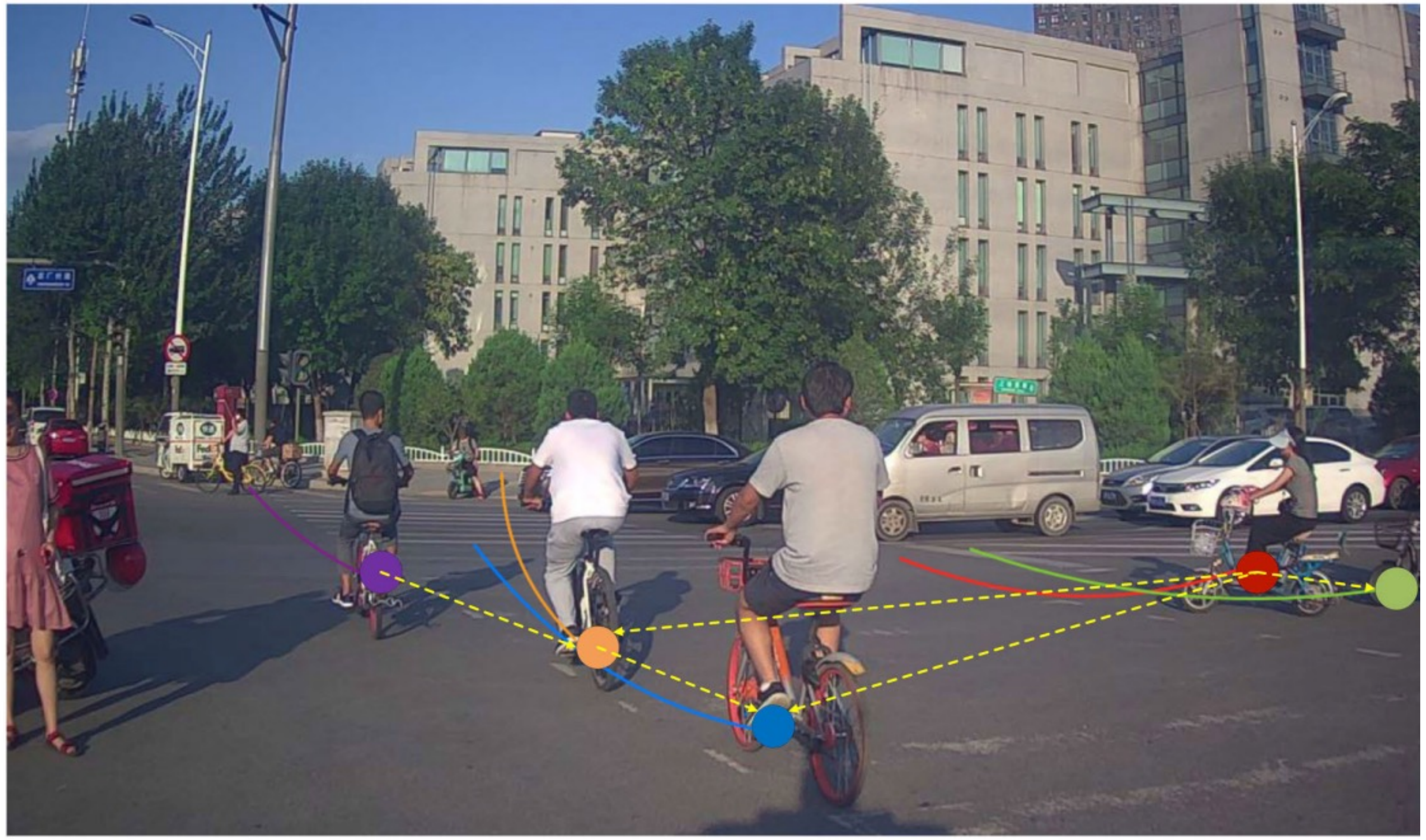
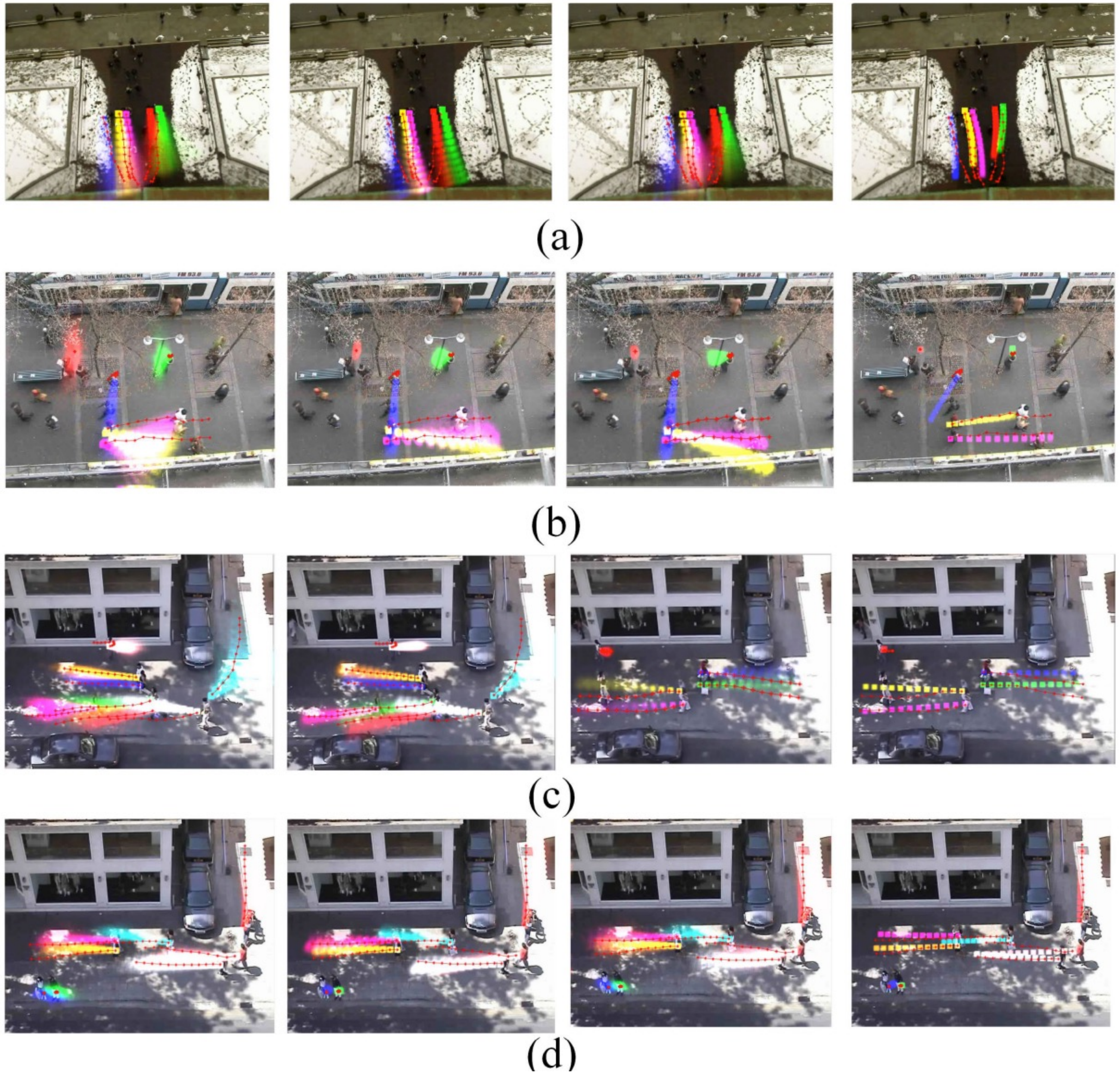


Fig. 1. Illustrations of trajectory prediction with captured social interacti
Dots of different colors represent the graph nodes that encode the mo
patterns of different traffic-agents. The dashed lines represent the graph ed
that capture the social interactions among different traffic-agents. The s
lines represent their future trajectories.



A Multi-Task Learning Network With a Collision-Aware Graph Transformer for Traffic-Agents Trajectory Prediction, 2024

DeepRob

Lecture 20
Video Processing
University of Michigan | Department of Robotics