Detailed structure of GNs-LSTM

Index	Input	Operation	Details
(0)	Task Vector	Task fea. encode	1) GRU(300, 64, batch_first=True, bidirectional=True), weights=((192, 300), (192, 64), (192,), (192,), (192, 300), (192, 64), (192,), (192,)) 2) Linear(1800, 640,), ReLU()
(1)	Scene graph X_E, X_S	Graph fea. encode1	1) Linear(267, 512), ReLU(), Linear(512, 11) 2) (node_mlp1): Linear(139, 512), ReLU(), Linear(512, 139) 3) (node_mlp2): Linear(267, 512), ReLU(), Linear(512, 128)
(2)	(1)	Graph fea. encode1	1) (edge_mlp1): Linear(267, 512), ReLU(), Linear(512, 11,) 2) (node_mlp1): Linear(139, 512), ReLU(), Linear(512, 139) 3) (node_mlp2): Linear(267, 512), ReLU(), Linear(512, 64)
(3)	(0), (2)	Sequence prediction	1) LSTM(22, 1280, batch_first=True), weights=((5120, 22), (5120, 1280), (5120,), (5120,))
(4)	(3)	Actions prediction	1) Linear(1280, 10), Softmax(), dropout=0.4
(5)	(3)	Objects prediction	1) Linear(1280, 11), Softmax(), dropout=0.4

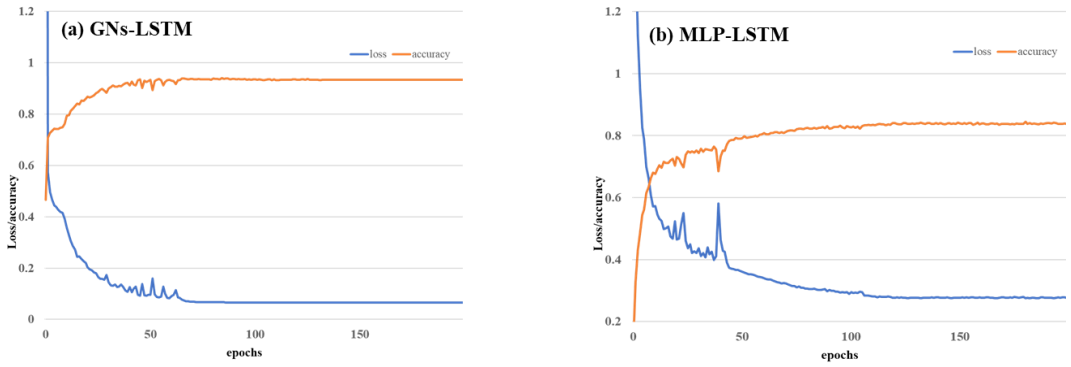
Detailed structure of MLP-LSTM

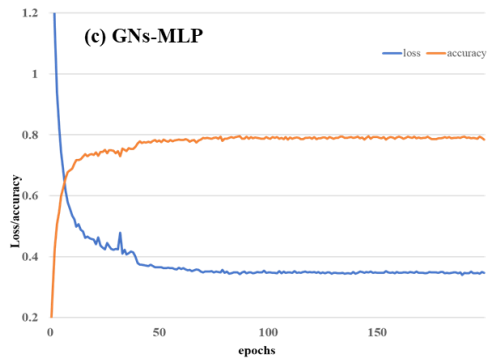
Index	Input	Operation	Details
(0)	Task Vector	Task fea. encode	1) GRU(300, 32, batch_first=True, bidirectional=True) weights=((96, 300), (96, 32), (96,), (96,), (96, 300), (96, 32), (96,), (96,)) 2) Linear(1800, 640,), ReLU()
(1)	Scene graph	Graph fea. encode	1) (rel_encode): Linear(10, 64), ReLU() 2) (att_encode): Linear(7, 64), ReLU() 3) (class_encode): GRU(300, 32, batch_first=True), weights=((96, 300), (96, 32), (96,), (96,), (96, 300), (96, 32), (96,), (96,))
(2)	(1)	Graph fea. encode	1) Linear(192, 64), ReLU()
(3)	(0), (2)	Sequence prediction	1) LSTM(22, 1280, batch_first=True), weights=((5120, 22), (5120, 1280), (5120,), (5120,))
(4)	(3)	Actions prediction	1) Linear(1280, 10), Softmax(), dropout=0.4
(5)	(3)	Objects prediction Eq. (5.2.2.5)	1) Linear(1280, 11), Softmax(), dropout=0.4

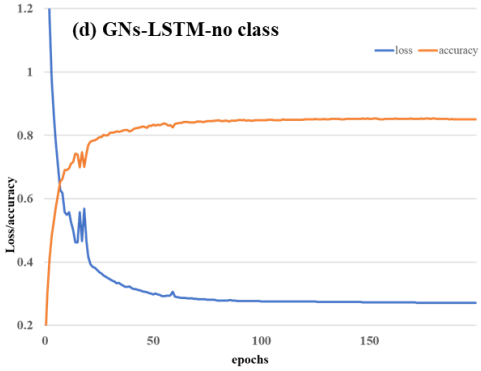
Detailed structure of GNs-MLP

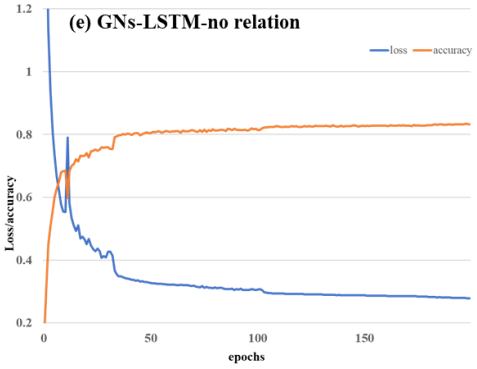
Index	Input	Operation	Details
(0)	Task Vector	Task fea. encode	1) GRU(300, 64, batch_first=True, bidirectional= True), weights=((192, 300), (192,64),(192,),(192,), (192, 300), (192, 64), (192,),(192,)) 2) Linear(1800, 640,), ReLU()
(1)	Scene graph X_E, X_S	Graph fea. encode1	1) (edge_mlp1):Linear(267, 512), ReLU(), Linear (512, 11) 2) (node_mlp1): Linear(139, 512), ReLU(), Linea r(512, 139) 3) (node_mlp2): Linear(267, 512), ReLU(), Linea r(512, 128)
(2)	(1)	Graph fea. encode1	1) (edge_mlp1): Linear(267, 512), ReLU(), Linea r(512, 11,) 2) (node_mlp1): Linear(139, 512), ReLU(), Linea r(512, 139) 3) (node_mlp2): Linear(267, 512), ReLU(), Linea r(512, 64)
(3)	(0), (2)	Seq._encoder	1) Linear(22, 640), ReLU()
(4)	(3)	Sequence prediction	1) Linear(1920, 1280), ReLU()
(5)	(4)	Actions prediction	1) Linear(1280, 10), Softmax(), dropout=0.4
(6)	(4)	Objects prediction	1) Linear(1280, 11), Softmax(), dropout=0.4

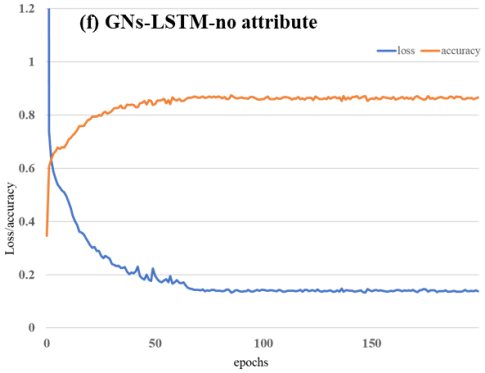
Figure 1 shows the change curves of loss and accuracy of each model in Table II during the test. It can be seen that with the progress of the learning process, the algorithm gradually summarizes the characteristics of the re-planning model and gets a high return on the target function value. After the 80th iteration, the oscillation tends to converge.




(c) GNs-MLP
loss
accuracy
Loss/accuracy
epochs
1.2
1
0.8
0.6
0.4
0.2
0
50
100
150


(d) GNs-LSTM-no class
loss
accuracy
Loss/accuracy
epochs
1.2
1
0.8
0.6
0.4
0.2
0
50
100
150


(e) GNs-LSTM-no relation
loss
accuracy
Loss/accuracy
epochs
1.2
1
0.8
0.6
0.4
0.2
0
50
100
150


(f) GNs-LSTM-no attribute
loss
accuracy
Loss/accuracy
epochs
1.2
1
0.8
0.6
0.4
0.2
0
50
100
150