

# DenseTNT

作者您好，很荣幸能和您进行一些沟通，我在复现源码跑出的结果和您论文指标非常一致，我相信这是一个非常好的代码方案，然后在重构您的代码的过程中自己有些疑问和不懂的地方，希望您能帮我解惑，再次感谢您！

目前我存在的一些问题：

1、关于loss 原文中loss是

$$\mathcal{L}_{S1} = \mathcal{L}_{\text{lane}} + \mathcal{L}_{\text{goal}} + \mathcal{L}_{\text{completion}}$$

但是在源码中 loss是不是只计算了 goal+completion ,关于lane的loss没找到对应于源码中的这个部分：

```
src > modeling > decoder.py > Decoder > goals_2D_per_example_calc_loss
169         labels_is_valid: List[np.ndarray]):
170     """
171     Calculate loss for a training example
172     """
173     final_idx = mapping[i].get('final_idx', -1)
174     gt_goal = gt_points[final_idx]
175     DE[i][final_idx] = np.sqrt((highest_goal[0] - gt_points[final_idx][0]) ** 2 + (highest_goal[1] - gt_points[final_idx][1]) ** 2)
176     if 'complete_traj' in args.other_params:
177         target_feature = self.goals_2D_mlp(torch.tensor(gt_points[final_idx], dtype=torch.float, device=device))
178         pass
179         if True:
180             target_feature.detach()
181             hidden_attention = self.complete_traj_cross_attention(
182                 target_feature.unsqueeze(0).unsqueeze(0), inputs[i][:inputs_lengths[i]].detach().unsqueeze(0).squeeze(
183                     0).squeeze(0)
184             )
185             predict_traj = self.complete_traj_decoder(
186                 torch.cat([hidden_states[i, 0, :].detach(), target_feature, hidden_attention], dim=-1)).view(
187                     [self.future_frame_num, 2])
188             loss[i] += (F.smooth_l1_loss(predict_traj, torch.tensor(gt_points, dtype=torch.float, device=device), reduction='none') * \
189                 torch.tensor(labels_is_valid[i], dtype=torch.float, device=device).view(self.future_frame_num, 1)).mean()
190
191     loss[i] += F.nll_loss(scores.unsqueeze(0),
192                         torch.tensor([mapping[i]['goals_2D_labels']], device=device))
```

2、lazy\_point 的问题，没有理解这个具体是干什么的？

```
src > modeling > decoder.py > Decoder > goals_2D_per_example_lazy_points
131     stage_one_topk = torch.topk(scores, k=stage_one_topk)
132     mapping[i]['stage_one_topk'] = stage_one_topk
133
134     return stage_one_topk_ids
135
136 def goals_2D_per_example_lazy_points(self, i, goals_2D, mapping, labels, device, scores,
137                                     get_scores_inputs, stage_one_topk_ids=None, gt_points=None):
138     if args.angoverse:
139         k = 150
140     else:
141         k = 40
142     _, topk_ids = torch.topk(scores, k=min(k, len(scores)))
143     topk_ids = topk_ids.tolist()
144
145     goals_2D_new = utils.get_neighbour_points(goals_2D[topk_ids], topk_ids=topk_ids, mapping=mapping[i])
146
147     goals_2D_new = torch.cat([torch.tensor(goals_2D_new, device=device, dtype=torch.float),
148                               torch.tensor(goals_2D, device=device, dtype=torch.float)], dim=0)
149
150     old_vector_num = len(goals_2D)
151
152     goals_2D = np.array(goals_2D_new.tolist())
153     # print('len', len(goals_2D))
```

```

154     scores = self.get_scores(goals_2D_new, *get_scores_inputs)
155
156     index = torch.argmax(scores).item()
157     point = np.array(goals_2D_new[index].tolist())
158
159     if not args.do_test:
160         label = np.array(labels[i]).reshape([self.future_frame_num, 2])
161         final_idx = mapping[i].get('final_idx', -1)
162         mapping[i]['goals_2D_labels'] = np.argmin(utils.get_dis(goals_2D, label[final_idx]))
163
164     return scores, point, goals_2D

```

3、关于轨迹生成部分，如果不使用set-predictor,如何来对heatmap的scores以及后续生成6条轨迹作出合理的解释

```

if 'complete_traj' in args.other_params:
    target_feature = self.goals_2D_mlp(torch.tensor(gt_points[final_idx], dtype=torch.float, device=device))
    pass
    if True:
        target_feature.detach_()
        hidden_attention = self.complete_traj_cross_attention(
            target_feature.unsqueeze(0).unsqueeze(0), inputs[i][:inputs_lengths[i]].detach().unsqueeze(0).squeeze(
                0).squeeze(0)
        )
        predict_traj = self.complete_traj_decoder(
            torch.cat([hidden_states[i, 0, :].detach(), target_feature, hidden_attention], dim=-1)).view(
                [self.future_frame_num, 2])

```

可能我提出的问题和对应代码位置有问题，请您帮忙指正！