TOPIC

3D Point Cloud Model 경량화

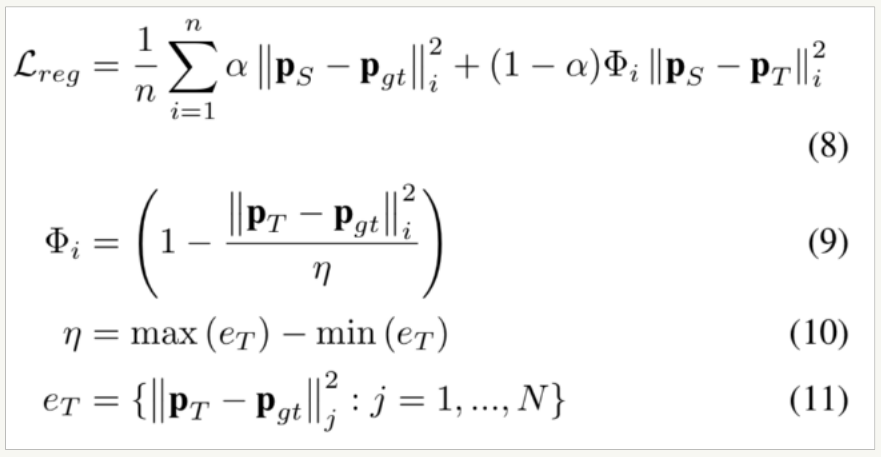
DATE

06.04

KEYPOINTS

NOTES

논문 Attentive Imitation Loss (AIL) Function 읽음.



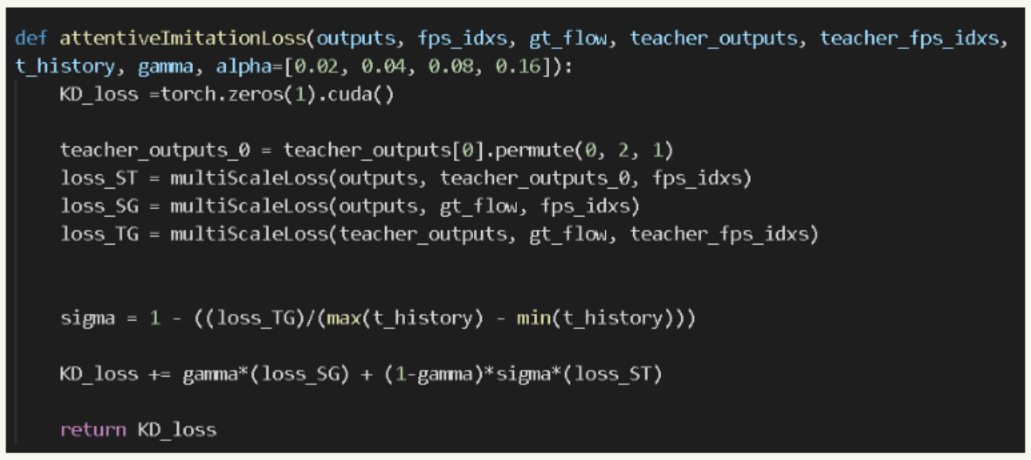
$$\mathcal{L}_{reg} = \frac{1}{n}\sum_{i=1}^{n} \alpha \left\|\mathbf{p}_S - \mathbf{p}_{gt}\right\|_i^2 + (1-\alpha)\Phi_i \left\|\mathbf{p}_S - \mathbf{p}_T\right\|_i^2 \tag{8}$$

$$\Phi_i = \left(1 - \frac{\left\|\mathbf{p}_T - \mathbf{p}_{gt}\right\|_i^2}{\eta}\right) \tag{9}$$

$$\eta = \max\left(e_T\right) - \min\left(e_T\right) \tag{10}$$

$$e_T = \left\{\left\|\mathbf{p}_T - \mathbf{p}_{gt}\right\|_j^2 : j = 1, \ldots, N\right\} \tag{11}$$

Loss Function 구현



```python
def attentiveImitationLoss(outputs, fps_idxs, gt_flow, teacher_outputs, teacher_fps_idxs,
t_history, gamma, alpha=[0.02, 0.04, 0.08, 0.16]):
    KD_loss =torch.zeros(1).cuda()

    teacher_outputs_0 = teacher_outputs[0].permute(0, 2, 1)
    loss_ST = multiScaleLoss(outputs, teacher_outputs_0, fps_idxs)
    loss_SG = multiScaleLoss(outputs, gt_flow, fps_idxs)
    loss_TG = multiScaleLoss(teacher_outputs, gt_flow, teacher_fps_idxs)


    sigma = 1 - ((loss_TG)/(max(t_history) - min(t_history)))

    KD_loss += gamma*(loss_SG) + (1-gamma)*sigma*(loss_ST)

    return KD_loss
```

기본 knowledge Distillation 보다
좋은 성능을 보여줌
Good

SUMMARY