

原代码中角度的计算方法如下

```
import torch
import torch.nn.functional as F
...
class RkdAngle(nn.Module):
    def forward(self, student, teacher):
        with torch.no_grad():
            td = (teacher.unsqueeze(0) - teacher.unsqueeze(1))
            norm_td = F.normalize(td, p=2, dim=2)
            t_angle = torch.bmm(norm_td, norm_td.transpose(1, 2)).view(-1)
            sd = (student.unsqueeze(0) - student.unsqueeze(1))
            norm_sd = F.normalize(sd, p=2, dim=2)
            s_angle = torch.bmm(norm_sd, norm_sd.transpose(1, 2)).view(-1)

            loss = F.smooth_l1_loss(s_angle, t_angle, reduction='mean')
        return loss
```

其中 teacher 是一个 64×512 的张量，来代表 64 个有 512 个变量的点

$$\text{teacher} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{64} \end{bmatrix}$$

$\text{td}=(\text{teacher}.\text{unsqueeze}(0) - \text{teacher}.\text{unsqueeze}(1))$ 会返回一个 $64 \times 64 \times 512$ 的张量，包含了 64 个点的全部以任意 2 点位一组的差（2 点可以为同一点，同时若 $x_i \neq x_j$ ， $x_i - x_j$ 和 $x_j - x_i$ 则视为 2 个不同的组）



$$td = \left[\begin{bmatrix} x_1 - x_1 \\ x_2 - x_1 \\ x_3 - x_1 \\ \vdots \\ x_{64} - x_1 \end{bmatrix}, \begin{bmatrix} x_1 - x_2 \\ x_2 - x_2 \\ x_3 - x_2 \\ \vdots \\ x_{64} - x_2 \end{bmatrix}, \dots, \begin{bmatrix} x_1 - x_{64} \\ x_2 - x_{64} \\ x_3 - x_{64} \\ \vdots \\ x_{64} - x_{64} \end{bmatrix} \right]$$

`norm_td = F.normalize(td, p=2, dim=2)` 会将所有向量单独归一化

$$\text{norm_td} = \left[\begin{bmatrix} [0] \\ \vdots \\ x_{64} - x_1 \\ \hline \|x_{64} - x_1\| \end{bmatrix}, \begin{bmatrix} \frac{x_1 - x_2}{\|x_1 - x_2\|} \\ \vdots \\ \frac{x_{64} - x_2}{\|x_{64} - x_2\|} \end{bmatrix}, \dots, \begin{bmatrix} \frac{x_1 - x_{64}}{\|x_1 - x_{64}\|} \\ \vdots \\ [0] \end{bmatrix} \right]$$

`torch.bmm(norm_sd, norm_sd.transpose(1, 2))` 将每一个 `norm_td` 里的二维张量与其自己的转置矩阵计算乘积，算出的结果就是排列组合之后全部可能的点积，

$$\left[\begin{bmatrix} [0] \\ \vdots \\ x_{64} - x_1 \\ \hline \|x_{64} - x_1\| \end{bmatrix} \begin{bmatrix} [0] \\ \vdots \\ x_{64} - x_1 \\ \hline \|x_{64} - x_1\| \end{bmatrix}^T, \dots, \begin{bmatrix} \frac{x_1 - x_{64}}{\|x_1 - x_{64}\|} \\ \vdots \\ [0] \end{bmatrix} \begin{bmatrix} \frac{x_1 - x_{64}}{\|x_1 - x_{64}\|} \\ \vdots \\ [0] \end{bmatrix}^T \right]$$

最后的 `view(-1)` 只是单独把所有的点积一个个列出来，生成一个 `list`。

学生的同理，最后两组数据计算 `loss`。所以原代码找点的逻辑是直接覆盖全部可能的组合。