# 6 Supplementary Material

## 6.1 Implementation of FCD

The implementation details of FCD are presented in Figure 4. Utilizing the output features from both the student and teacher models, denoted as $F_s$ and $F_t$ respectively, our method ensures ease of implementation.

```
import torch.nn as nn
from torch.nn import functional as F

def cosine_similarity(x, y, eps=1e-8):
    return (x * y).sum(1) / (x.norm(dim=1) * y.norm(dim=1) + eps)

def pearson_correlation(x, y, eps=1e-8):
    return cosine_similarity(x - x.mean(-1).unsqueeze(-1),
                             y - y.mean(-1).unsqueeze(-1), eps)

def plc_loss(x, y):
    return 1 - pearson_correlation(x, y).mean()

def token_relation_loss(y_s, y_t):
    R_s = y_s.bmm(y_s.transpose(-1, -2))
    R_t = y_t.bmm(y_t.transpose(-1, -2))

    R_s = R_s.view(B, -1)
    R_t = R_t.view(B, -1)
    return plc_loss(R_s, R_t)

def sample_relation_loss(y_s, y_t):
    y_s = y_s.permute(1, 0, 2)
    y_t = y_t.permute(1, 0, 2)
    return token_relation_loss(y_s, y_t)

class FCDLoss(nn.Module):
    def __init__(self, alpha, beta):
        super(FCDLoss, self).__init__()
        self.alpha = alpha
        self.beta = beta

    def forward(self, F_s, F_t):
        B, N, D = F_t.shape
        F_s = F.normalize(F_s, dim=-1)
        F_t = F.normalize(F_t, dim=-1)
        loss_token = token_relation_loss(F_s, F_t)
        loss_sample = sample_relation_loss(F_s, F_t)
        kd_loss = self.alpha * loss_token + self.beta * loss_sample
        return kd_loss
```

Figure 4: The PyTorch implementation of FCD.

## 6.2 Proofs

**Invariance of Pearson's Correlation under Positive Linear Transformation.** Let's consider two random variables $X$ and $Y$. A positive linear transformation on $X$ and $Y$ can be formulated as $X' = aX + b$ and $Y' = cY + d$, where $a \times c > 0$ and $b$, $d$ are arbitrary constants. Applying these transformations to the means of $X$ and $Y$, we derive $\mu_{X'} = a\mu_X + b$ and $\mu_{Y'} = c\mu_Y + d$. By

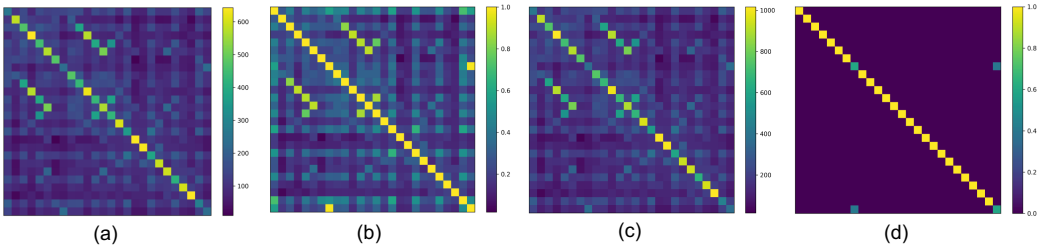


Figure 5: Visualization of the token-level relational features under different normalization functions. (a) Pre-Norm. (b) $\ell_2$-Norm. (c) Layer-Norm. (d) Softmax-Norm.

substituting the transformed variables and their corresponding means into Equation 6:

$$\begin{aligned}\rho'(X,Y) &= \frac{\sum((aX_i+b)-(a\mu_X+b))((cY_i+d)-(c\mu_Y+d))}{\sqrt{\sum((aX_i+b)-(a\mu_X+b))^2}\sqrt{\sum((cY_i+d)-(c\mu_Y+d))^2}} \\ &= \frac{\sum(a(X_i-\mu_X))(c(Y_i-\mu_Y))}{(a\sqrt{\sum(X_i-\mu_X)^2})(a\sqrt{\sum(Y_i-\mu_Y)^2})} = \rho(X,Y)\end{aligned} \tag{14}$$

**Relationship among PLC, KL divergence and MSE** In Equation 11, $\phi(\cdot)$ represents the softmax function, while $\tau$ denotes a temperature parameter controlling the softness of the distributions.

$$\frac{\partial\,\mathcal{L}_{KL}}{\partial\hat{\mathcal{F}}_S^i} = \tau\left(\phi_s(\tau)-\phi_t(\tau)\right) = \tau\left(\frac{\exp(\hat{\mathcal{F}}_S^i/\tau)}{\sum_{j=1}^m\exp(\hat{\mathcal{F}}_S^j/\tau)} - \frac{\exp(\hat{\mathcal{F}}_T^i/\tau)}{\sum_{j=1}^m\exp(\hat{\mathcal{F}}_T^j/\tau)}\right), \tag{15}$$

Assuming $\tau$ is significantly large compared to the magnitude of the normalized features and both $\hat{\mathcal{F}}_S^i$ and $\hat{\mathcal{F}}_T^i$ are drawn from a standard normal distribution. In this case, the term $\hat{\mathcal{F}}^i/\tau$ becomes quite small, allowing us to approximate $\exp(\hat{\mathcal{F}}^i/\tau)$ as $1+\hat{\mathcal{F}}^i/\tau$. This simplification leads to an approximation of the gradient in Equation 15:

$$\frac{\partial\mathcal{L}_{KL}}{\partial\hat{\mathcal{F}}_S^i} \approx \tau\left(\frac{1+\hat{\mathcal{F}}_S^i/\tau}{M+\sum_{j=1}\hat{\mathcal{F}}_S^j/\tau} - \frac{1+\hat{\mathcal{F}}_T^i/\tau}{M+\sum_j\hat{\mathcal{F}}_T^j/\tau}\right) \tag{16}$$

Given that the sums $\sum_j\hat{\mathcal{F}}_S^j$ and $\sum_j\hat{\mathcal{F}}_T^j$ are both zero, Equation 16 simplifies further to:

$$\frac{\partial\mathcal{L}_{KL}}{\partial\hat{\mathcal{F}}_S^i} = \frac{1}{M}(\hat{\mathcal{F}}_S-\hat{\mathcal{F}}_T) = \frac{\partial\mathcal{L}_{MSE}}{\partial\hat{\mathcal{F}}_S^i} \tag{17}$$

Moreover, considering $\frac{1}{m-1}\sum_i\hat{\mathcal{F}}_S^2 = 1$ and $\frac{1}{m-1}\sum_i\hat{\mathcal{F}}_T^2 = 1$, we can reformulate the MSE as follows:

$$\begin{aligned}\mathcal{L}_{MSE}(\hat{\mathcal{F}}_S,\hat{\mathcal{F}}_T) &= \frac{1}{2M}\sum(\hat{\mathcal{F}}_S-\hat{\mathcal{F}}_T)^2 \\ &= \frac{1}{2M}\left((2M-2)-2\sum_{i=1}^m\hat{\mathcal{F}}_S\hat{\mathcal{F}}_T\right) \\ &= \frac{2M-2}{2M}(1-\rho(\mathcal{F}_S,\mathcal{F}_T)) \approx \mathcal{L}_{PLC}(\mathcal{F}_S,\mathcal{F}_T)\end{aligned} \tag{18}$$

Thus, we demonstrate that minimizing KL divergence between normalized features under a high-temperature limit is equivalent to minimizing the MSE between normalized ones, which is in turn equivalent to maximizing the PLC between the original features.

### 6.3 More Experiments Results

**Effect of different Normalization and Loss Functions** Section 3.4 in the main text clarifies the intrinsic relationship between KL divergence, MSE, and PLC. However, the assumption that normalized features follow a Gaussian distribution may not invariably be valid. To investigate the

performance of varying normalization and loss functions, we conducted a series of experiments, setting the temperature $\tau$ to 10 when utilizing KL divergence as the loss function. Table 4 demonstrates that the $\ell 2$ norm consistently outperforms other normalization functions. Minimizing KL divergence between layer-normalized features in the high-temperature limit can yield results comparable to MSE and PLC. To further underscore the benefits of $\ell 2$ normalization, we provide a visualization of the pre-normalized and post-normalized token-level relational features in Figure 5. In contrast to $\ell 2$ normalization, other functions often produce a wider range with larger values, suggesting that directly imitating these normalized features could introduce significant noise, potentially leading to subpar results.

Table 4: Results of the loss function combined with different normalization mechanism.

| Scheme | MNLI-m | QQP | CoLA | Average |
|---|---|---|---|---|
| LayerNorm + KL | 83.4 | 71.5 | 51.4 | 68.8 |
| LayerNorm + MSE | 83.6 | 71.8 | 51.5 | 69.0 |
| PreNorm + PLC | 83.3 | 71.2 | 51.7 | 68.7 |
| Softmax + PLC | 83.4 | 71.7 | 51.5 | 68.9 |
| $\ell_2$ + PLC | 83.8 | 72.0 | 52.0 | 69.3 |

**Results on SQuAD v1.1 and v2.0.** To further demonstrate FCD's effectiveness, we applied it to the question-answering tasks of SQuAD v1.1 Rajpurkar et al. [2016] and SQuAD v2.0 Rajpurkar et al. [2018]. We framed these tasks as sequence labeling problems, predicting the likelihood of each token being the start or end of an answer span. We employed the F1 metric for both versions of SQuAD. $\text{BERT}_{\text{BASE}}$ was used as the teacher model, and a $6 \times 768$ model served as the student model. The results, presented in Table 5, indicate that FCD can enhance the student model's performance on both tasks.

Table 5: Results of baselines and FCD on question answering tasks.

| Method | SQuAD 1.1 | SQuAD 2.0 |
|---|---|---|
| $\text{BERT}_{\text{BASE}}$ | 88.7 | 78.8 |
| $\text{DistilBERT}_6$ | 86.2 | 69.5 |
| $\text{TinyBERT}_6$ | 87.5 | 77.7 |
| **Ours** | 88.2 | 78.4 |

### 6.4 Discussion

**Limitations.** While FCD demonstrates consistent performance improvements across diverse Transformer-based models, its effectiveness may be less pronounced on other architectures such as Recurrent Neural Networks (RNNs). The feature relationships in RNNs are not as explicit as in Transformers, potentially limiting the applicability and impact of FCD.

**Societal impacts.** The extensive computational resources required to evaluate our proposed method could significantly contribute to carbon emissions, thereby raising sustainability concerns. However, the objective of our approach is to enhance the efficiency of lightweight models through knowledge distillation. This enhancement could ultimately replace heavier models in production settings, resulting in substantial energy savings. Thus, the thorough validation of FCD's efficacy is a necessary trade-off to ensure its potential benefits.