

## A CROSS ENTROPY AND KL-DIVERGENCE

Let  $\mathcal{S}(x)$  and  $\mathcal{T}(x)$  to be the softmax outputs of the substitute model and the target model. The expressions of the KL-divergence loss and the cross entropy loss are shown in the following.

$$\mathcal{L}_{\text{KL}}(x) = \sum_{i=1}^N \mathcal{T}_i(x) \log \left( \frac{\mathcal{T}_i(x)}{\mathcal{S}_i(x)} \right)$$

$$\mathcal{L}_{\text{CE}}(x) = - \sum_{i=1}^N \mathcal{T}_i(x) \log \mathcal{S}_i(x)$$

In the black-box setting of this paper,  $\mathcal{T}$ 's model parameters are unavailable. Therefore  $\mathcal{T}(x)$  is a constant vector. In this case, we have

$$\nabla_x \mathcal{L}_{\text{KL}}(x) = \nabla_x \mathcal{L}_{\text{CE}}(x) = - \sum_{i=1}^N \mathcal{T}_i(x) \nabla [\log \mathcal{S}_i(x)]$$

But an interesting observation is that in white-box settings ( $\mathcal{T}$  is derivable) cross entropy won't suffer from vanishing gradients but KL-divergence can suffer from vanishing gradients. The justification is in the following. Taking the gradient over  $\mathcal{L}_{\text{KL}}(x)$ , we have

$$\begin{aligned} \nabla_x \mathcal{L}_{\text{KL}}(x) &= \sum_{i=1}^N \frac{\partial \mathcal{T}_i}{\partial x} \log \frac{\mathcal{T}_i}{\mathcal{S}_i} + \frac{\partial \mathcal{T}_i}{\partial x} - \frac{\partial \mathcal{S}_i}{\partial x} \frac{\mathcal{T}_i}{\mathcal{S}_i} \\ &= \sum_{i=1}^N \frac{\partial \mathcal{T}_i}{\partial x} \log \frac{\mathcal{T}_i}{\mathcal{S}_i} - \frac{\partial \mathcal{S}_i}{\partial x} \frac{\mathcal{T}_i}{\mathcal{S}_i}, \end{aligned}$$

where  $\sum_{i=1}^N \frac{\partial \mathcal{T}_i}{\partial x} = 0$  because  $\sum_{i=1}^N \mathcal{T}_i = 1$ . When  $\mathcal{S}$  converges to  $\mathcal{T}$  we have  $\mathcal{T}_i(x) = \mathcal{S}_i(x) (1 + \epsilon_i(x))$  where  $\epsilon_i(x)$  tends to 0 during the convergence process. When  $\epsilon_i(x)$  tends to 0,  $\log(1 + \epsilon_i(x)) \approx \epsilon_i(x)$ . Then we have

$$\begin{aligned} \nabla_x \mathcal{L}_{\text{KL}}(x) &\approx \sum_{i=1}^N \frac{\partial \mathcal{T}_i}{\partial x} \epsilon_i - \frac{\partial \mathcal{S}_i}{\partial x} (1 + \epsilon_i) \\ &\approx \sum_{i=1}^N \epsilon_i \left( \frac{\partial \mathcal{T}_i}{\partial x} - \frac{\partial \mathcal{S}_i}{\partial x} \right), \end{aligned} \tag{3}$$

where we have applied  $\sum_{i=1}^N \frac{\partial \mathcal{S}_i}{\partial x} = 0$  because  $\sum_{i=1}^N \mathcal{S}_i = 1$ . According to Equation (3), the gradient of  $\mathcal{L}_{\text{KL}}(x)$  will gradually vanish after many iterations. Then taking the gradient over  $\mathcal{L}_{\text{CE}}(x)$ , we have

$$\begin{aligned} \nabla_x \mathcal{L}_{\text{CE}}(x) &= - \sum_{i=1}^N \frac{\partial \mathcal{T}_i}{\partial x} \log \mathcal{S}_i + \frac{\partial \mathcal{S}_i}{\partial x} \frac{\mathcal{T}_i}{\mathcal{S}_i} \\ &\approx - \sum_{i=1}^N \frac{\partial \mathcal{T}_i}{\partial x} \log \mathcal{S}_i - \sum_{i=1}^N \frac{\partial \mathcal{S}_i}{\partial x} - \sum_{i=1}^N \epsilon_i \frac{\partial \mathcal{S}_i}{\partial x} \\ &= - \sum_{i=1}^N \frac{\partial \mathcal{T}_i}{\partial x} \log \mathcal{S}_i - \sum_{i=1}^N \epsilon_i \frac{\partial \mathcal{S}_i}{\partial x} \end{aligned}$$

where the first term won't vanish during the iterations. Therefore the cross entropy loss won't suffer from vanishing gradients. Note that we have applied  $\sum_{i=1}^N \frac{\partial \mathcal{S}_i}{\partial x} = 0$ .

## B PROOF OF THEOREM 1

### B.1 NOTATIONS AND LEMMAS

For simplicity, we use  $\mathcal{T}(z; \theta_g^{(t)})$  to denote  $\mathcal{T}(\mathcal{G}(z; \theta_g^{(t)}))$  and  $\mathcal{S}(z; \theta_g^{(t)}, \theta_s^{(t)})$  to denote  $\mathcal{S}(\mathcal{G}(z; \theta_g^{(t)}); \theta_s^{(t)})$ . In order to prove Theorem 1, we firstly derive Lemma 1 according to the assumptions in Section 3.3.

**Lemma 1.** *After the training of  $\mathcal{G}$  (line 11-14, Algorithm 1) in round  $t$ , given  $z \in Z$ , we have  $\text{CE}(\mathcal{T}(z; \theta_g^{(t)}), \mathcal{S}(z; \theta_g^{(t)}, \theta_s^{(t)})) \leq \text{CE}(\mathcal{T}(z; \theta_g^{(t-1)}), \mathcal{S}(z; \theta_g^{(t-1)}, \theta_s^{(t)}))$ .*

*Proof.* After training  $\mathcal{G}$  we have

$$\begin{aligned}\mathcal{S}_{i^*}(z, \theta_g^{(t)}, \theta_s^{(t)}) &\geq \mathcal{S}_{i^*}(z, \theta_g^{(t-1)}, \theta_s^{(t)}), \\ \mathcal{T}_{i^*}(z, \theta_g^{(t)}, \theta_s^{(t)}) &\geq \mathcal{T}_{i^*}(z, \theta_g^{(t-1)}, \theta_s^{(t)}),\end{aligned}$$

where  $i^* = \arg \max_i \mathcal{S}_i(\mathcal{G}(z; \theta_g^{(t-1)}); \theta_s^{(t)}) = \arg \max_i \mathcal{T}_i(\mathcal{G}(z; \theta_g^{(t-1)}))$ . Then we have

$$\begin{aligned}\text{CE}(\mathcal{T}(z; \theta_g^{(t)}), \mathcal{S}(z; \theta_g^{(t)}, \theta_s^{(t)})) &= - \sum_{i=1}^N \mathcal{T}_i(z; \theta_g^{(t)}) \log \mathcal{S}_i(z; \theta_g^{(t)}, \theta_s^{(t)}) \\ &\leq - \sum_{i=1}^N \mathcal{T}_i(z; \theta_g^{(t-1)}) \log \mathcal{S}_i(z; \theta_g^{(t)}, \theta_s^{(t)}) \\ &\leq - \sum_{i=1}^N \mathcal{T}_i(z; \theta_g^{(t-1)}) \log \mathcal{S}_i(z; \theta_g^{(t-1)}, \theta_s^{(t)}) \\ &= \text{CE}(\mathcal{T}(z; \theta_g^{(t-1)}), \mathcal{S}(z; \theta_g^{(t-1)}, \theta_s^{(t)}))\end{aligned}$$

### B.2 COMPLETING THE PROOF

**Theorem 1.** *Given  $z \in Z$ . Let  $f(\theta_s^{(t)}) = \text{CE}(\mathcal{T}(\mathcal{G}(z; \theta_g^{(t)})), \mathcal{S}(\mathcal{G}(z; \theta_g^{(t)}); \theta_s^{(t)}))$ . Training the substitute model by Algorithm 1, we have  $\lim_{t \rightarrow \infty} f(\theta_s^{(t)}) = \epsilon^*$ , where  $\epsilon^* \geq 0$ .*

*Proof.* We can simplify  $f(\theta_s^{(t)})$  as

$$f(\theta_s^{(t)}) = \text{CE}(\mathcal{T}(z; \theta_g^{(t)}), \mathcal{S}(z; \theta_g^{(t)}, \theta_s^{(t)})),$$

where  $t$  is used to index the training rounds. Using Lemma 1, We have

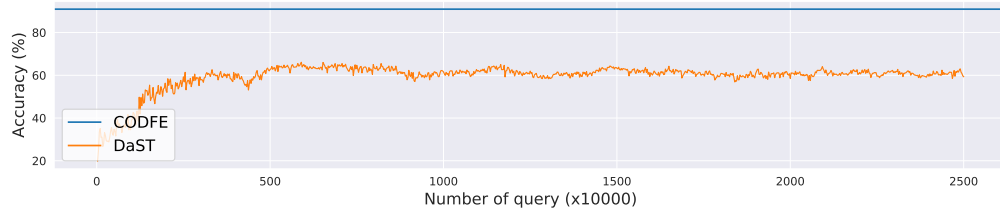
$$f(\theta_s^{(t+1)}) \leq \text{CE}(\mathcal{T}(z; \theta_g^{(t)}), \mathcal{S}(z; \theta_g^{(t)}, \theta_s^{(t+1)})).$$

Since the cross entropy loss is the loss function of  $\mathcal{S}$ , we have

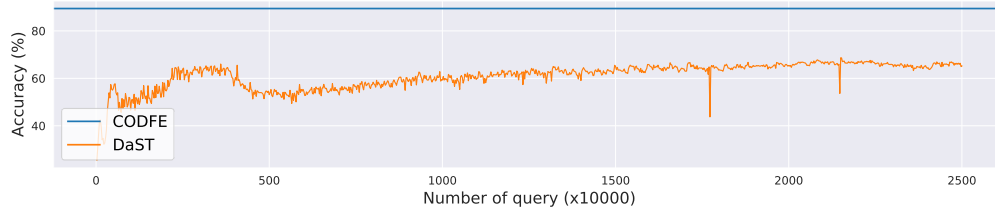
$$\begin{aligned}f(\theta_s^{(t+1)}) &\leq \text{CE}(\mathcal{T}(z, \theta_g^{(t)}), \mathcal{S}(z, \theta_g^{(t)}, \theta_s^{(t+1)})) \\ &\leq \text{CE}(\mathcal{T}(z, \theta_g^{(t)}), \mathcal{S}(z, \theta_g^{(t)}, \theta_s^{(t)})) \\ &= f(\theta_s^{(t)})\end{aligned}$$

Therefore, we know that  $f(\theta_s)$  is monotone decreasing during the training.  $f(\theta_s) = 0$  if and only if  $\mathcal{T}(z; \theta_g^{(t)}) = \mathcal{S}(z; \theta_g^{(t)}, \theta_s^{(t)})$ . Otherwise  $f(\theta_s) > 0$ . Since  $f(\theta_s) \geq 0$ , it will converge. However the outputs of  $\mathcal{S}$  and  $\mathcal{T}$  usually won't be exactly the same. Then the convergence can be formally represented as  $\lim_{t \rightarrow \infty} f(\theta_s^{(t)}) = \epsilon^*$ , where  $\epsilon^* \geq 0$ .

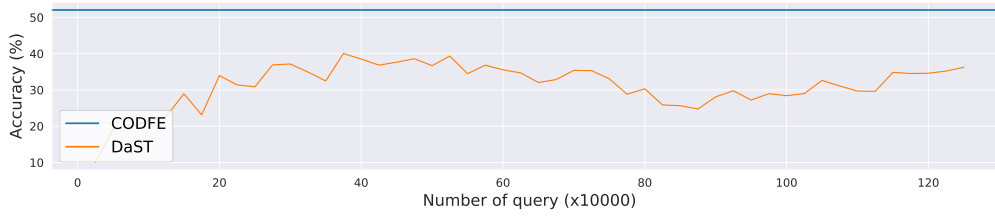
## C ADDITIONAL EXPERIMENTAL RESULTS



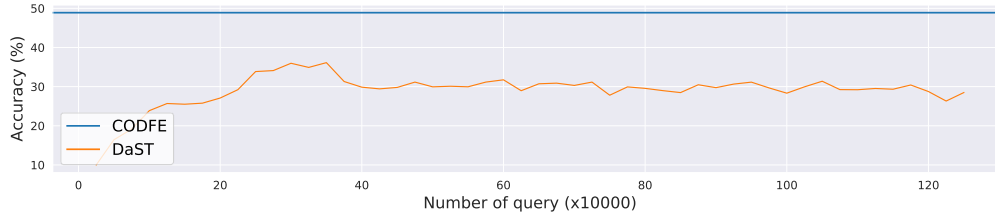
(a) Probability-only:MNIST



(b) Label-only:MNIST



(c) Label-only:Fashion-MNIST



(d) Probability-only:Fashion-MNIST

Figure 4: Substitute model accuracy.