A Proofs

A.1 Proof of Stein's identity

We recap the proof of Stein's identity following He *et al.* [34], where

$$\begin{aligned}
\nabla_{\mathbf{W}}\mathcal{L}_\sigma(\mathbf{W};\mathbb{D}) &= \nabla_{\mathbf{W}}\mathbb{E}_{\boldsymbol{\delta}\sim\mathcal{N}(0,\sigma^2 I)}\mathcal{L}(\mathbf{W}+\boldsymbol{\delta};\mathbb{D}) \\
&= (2\pi)^{-\frac{n}{2}}\cdot\nabla_{\mathbf{W}}\int\mathcal{L}(\mathbf{W}+\boldsymbol{\delta};\mathbb{D})\cdot\exp\left(-\frac{\|\boldsymbol{\delta}\|_2^2}{2\sigma^2}\right)d\boldsymbol{\delta} \\
&= (2\pi)^{-\frac{n}{2}}\cdot\int\mathcal{L}(\widetilde{\mathbf{W}};\mathbb{D})\cdot\nabla_{\mathbf{W}}\exp\left(-\frac{\|\widetilde{\mathbf{W}}-\mathbf{W}\|_2^2}{2\sigma^2}\right)d\widetilde{\mathbf{W}} \\
&= (2\pi)^{-\frac{n}{2}}\cdot\int\mathcal{L}(\mathbf{W}+\boldsymbol{\delta};\mathbb{D})\cdot\frac{\boldsymbol{\delta}}{\sigma^2}\cdot\exp\left(-\frac{\|\boldsymbol{\delta}\|_2^2}{2\sigma^2}\right)d\boldsymbol{\delta} \\
&= \mathbb{E}_{\boldsymbol{\delta}\sim\mathcal{N}(0,\sigma^2\mathbf{I})}\left[\frac{\boldsymbol{\delta}}{\sigma^2}\mathcal{L}(\mathbf{W}+\boldsymbol{\delta};\mathbb{D})\right].
\end{aligned} \tag{9}$$

By symmetry, we change $\boldsymbol{\delta}$ to $-\boldsymbol{\delta}$ and obtain

$$\nabla_{\mathbf{W}}\mathcal{L}_\sigma(\mathbf{W};\mathbb{D}) = -\mathbb{E}_{\boldsymbol{\delta}\sim\mathcal{N}(0,\sigma^2\mathbf{I})}\left[\frac{\boldsymbol{\delta}}{\sigma^2}\mathcal{L}(\mathbf{W}-\boldsymbol{\delta};\mathbb{D})\right], \tag{10}$$

and further we prove that

$$\begin{aligned}
\nabla_{\mathbf{W}}\mathcal{L}_\sigma(\mathbf{W};\mathbb{D}) &= \frac{1}{2}\mathbb{E}_{\boldsymbol{\delta}\sim\mathcal{N}(0,\sigma^2\mathbf{I})}\left[\frac{\boldsymbol{\delta}}{\sigma^2}\mathcal{L}(\mathbf{W}+\boldsymbol{\delta};\mathbb{D})\right] - \frac{1}{2}\mathbb{E}_{\boldsymbol{\delta}\sim\mathcal{N}(0,\sigma^2\mathbf{I})}\left[\frac{\boldsymbol{\delta}}{\sigma^2}\mathcal{L}(\mathbf{W}-\boldsymbol{\delta};\mathbb{D})\right] \\
&= \mathbb{E}_{\boldsymbol{\delta}\sim\mathcal{N}(0,\sigma^2\mathbf{I})}\left[\frac{\boldsymbol{\delta}}{2\sigma^2}\Delta\mathcal{L}(\mathbf{W},\boldsymbol{\delta};\mathbb{D})\right].
\end{aligned}$$

□

A.2 Proof of Theorem 1

We rewrite the format of $\widehat{\nabla_{\mathbf{W}}}\mathcal{L}(\mathbf{W};\mathbb{D})$ as follows:

$$\begin{aligned}
\widehat{\nabla_{\mathbf{W}}}\mathcal{L}(\mathbf{W};\mathbb{D}) &= \frac{1}{K}\sum_{k=1}^{K}[\frac{\boldsymbol{\delta}_k}{2\sigma^2}\Delta\mathcal{L}(\mathbf{W},\boldsymbol{\delta}_k;\mathbb{D})] \\
&= \frac{1}{K}\sum_{k=1}^{K}[\frac{\boldsymbol{\delta}_k}{2\sigma^2}\left(2\boldsymbol{\delta}_k^\top\nabla_{\mathbf{W}}\mathcal{L}(\mathbf{W};\mathbb{D})+o(\|\boldsymbol{\delta}_k\|_2^2)\right)] \quad \text{(using Eq. (2))} \\
&= \frac{1}{K\sigma^2}\sum_{k=1}^{K}[\boldsymbol{\delta}_k\boldsymbol{\delta}_k^\top]\nabla_{\mathbf{W}}\mathcal{L}(\mathbf{W};\mathbb{D}) + \frac{1}{K}\sum_{k=1}^{K}\frac{\boldsymbol{\delta}_k}{2\sigma^2}o(\|\boldsymbol{\delta}_k\|_2^2) \\
&= \widehat{\boldsymbol{\Sigma}}\nabla_{\mathbf{W}}\mathcal{L}(\mathbf{W};\mathbb{D}) + \frac{1}{K}\sum_{k=1}^{K}\frac{\boldsymbol{\delta}_k}{2\sigma^2}o(\|\boldsymbol{\delta}_k\|_2^2).
\end{aligned} \tag{11}$$

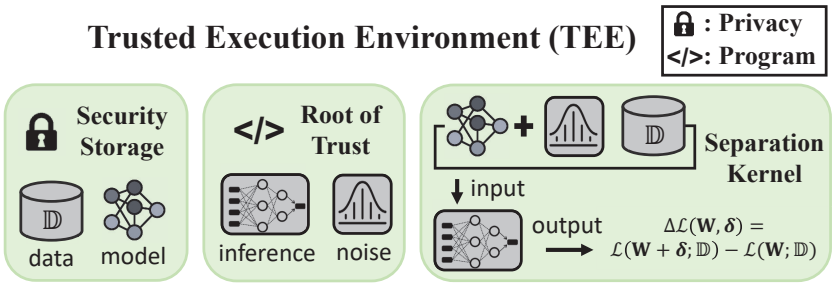

Fig. 4: A sketch map to run BAFFLE in one trusted execution environment. The pipeline contains three steps: (1) Load the data and model into the security storage. (2) Load the code of BAFFLE into the root of trust. (3) Run the BAFFLE program in a separation kernel.

Then we prove $\frac{1}{K}\sum_{k=1}^{K}\frac{\boldsymbol{\delta}_k}{2\sigma^2}o(\|\boldsymbol{\delta}_k\|_2^2)=o(\widehat{\boldsymbol{\delta}})$. Suppose $\boldsymbol{\delta}_k=(\delta_{k,1},\cdots,\delta_{k,n})$, then we have $\frac{\|\boldsymbol{\delta}_k\|_2^2}{\sigma^2}=\sum_{i=1}^{n}(\frac{\delta_{k,i}}{\sigma})^2$. Since $\forall i, \frac{\delta_{k,i}}{\sigma}\sim\mathcal{N}(0,1)$, we have $\frac{\|\boldsymbol{\delta}_k\|_2^2}{\sigma^2}\sim\chi^2(n)$ and $\mathbb{E}(\frac{\|\boldsymbol{\delta}_k\|_2^2}{\sigma^2})=n$. So with high probability, $\frac{o(\|\boldsymbol{\delta}_k\|_2^2)}{\sigma^2}=o(n)$. Substituting it into Eq. (11), we have with high probability,

$$\frac{1}{K}\sum_{k=1}^{K}\frac{\boldsymbol{\delta}_k}{2\sigma^2}o(\|\boldsymbol{\delta}_k\|_2^2)=\widehat{\boldsymbol{\delta}}\cdot o(n)=o(\widehat{\boldsymbol{\delta}}),$$

where we regard n as a constant for a given model architecture. Finally, we prove $\mathbb{E}[\widehat{\boldsymbol{\delta}}]=\mathbf{0}$ and $\mathbb{E}[\widehat{\boldsymbol{\Sigma}}]=\mathbf{I}$. It is trivial that $\mathbb{E}[\widehat{\boldsymbol{\delta}}]=\mathbf{0}$ since $\widehat{\boldsymbol{\delta}}\sim\mathcal{N}(0,\frac{1}{K\sigma^2}\mathbf{I})$. For $\mathbb{E}[\widehat{\boldsymbol{\Sigma}}]=\mathbf{I}$, we can observe by examining each of its entries

$$\widehat{\boldsymbol{\Sigma}}_{[ij]}=\frac{1}{K\sigma^2}\sum_{k=1}^{K}\delta_{k[i]}\delta_{k[j]}=\frac{1}{K}\sum_{k=1}^{K}\frac{\delta_{k[i]}}{\sigma}\frac{\delta_{k[j]}}{\sigma},\tag{12}$$

where we have used subscripts $[ij]$ and $[i]$ to denote the usual indexing of matrices and vectors. Specifically, for diagonal entries (i.e., $i=j$), we observe $K\cdot\widehat{\boldsymbol{\Sigma}}_{[ii]}=\sum_{k=1}^{K}\left(\frac{\delta_{k[i]}}{\sigma}\right)^2$ distributes as $\chi^2(K)$, which means $\mathbb{E}[\widehat{\boldsymbol{\Sigma}}_{[ii]}]=1=\mathbf{I}_{[ii]}$ and $\mathrm{Var}[\widehat{\boldsymbol{\Sigma}}_{[ii]}]=\frac{2}{K}$; for non-diagonal entries (i.e., $i\neq j$), we have $\mathbb{E}[\widehat{\boldsymbol{\Sigma}}_{[ij]}]=\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\frac{\delta_{k[i]}}{\sigma}\frac{\delta_{k[j]}}{\sigma}\right]=\frac{1}{K}\sum_{k=1}^{K}\frac{\mathbb{E}[\delta_{k[i]}]}{\sigma}\frac{\mathbb{E}[\delta_{k[j]}]}{\sigma}=0=\mathbf{I}_{[ij]}$, due to the independence between different dimensions in $\boldsymbol{\delta}_k$. □

B Trusted execution environment

A trusted execution environment (TEE) [75] is regarded as the ultimate solution for defending against all white-box attacks by preventing any model exposure. TEE protects both data and model security with three components: physical secure storage to ensure the confidentiality, integrity, and tamper-resistance of stored data; a root of trust to load trusted code; and a separate kernel to execute

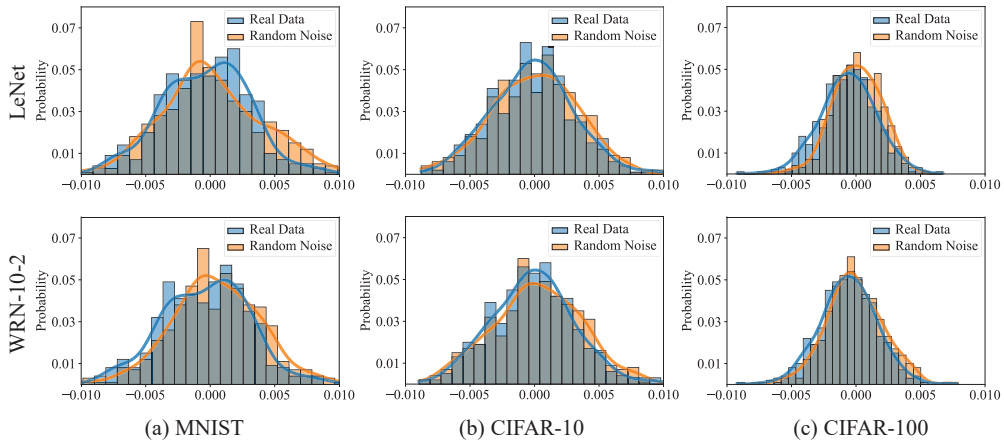

Fig. 5: The robustness of BAFFLE to inference attacks. For real data, we randomly sample some input-label pairs from the validation dataset. For random noise, we generate input-label pairs from standard normal distribution. We sample 500 perturbations $\boldsymbol{\delta}$ from $\mathcal{N}(0, \sigma^2\mathbf{I})$, collect the values of $\Delta\mathcal{L}(\mathbf{W}, \boldsymbol{\delta}; \mathbb{D})$ for real data and random noise separately, and compare their distributions.

code in an isolated environment, as illustrated in Figure 4. Using TEE, the FL system is able to train deep models without revealing model specifics. However, due to the security guarantee, the usable memory of TEE is typically small [87] (e.g., 90MB on Intel SGX for Skylake CPU [61]), which is considerably less than what deep models require for backpropagation (e.g., $\geq$ 5GB for VGG-16 [24]).

C Convergence analyses of deep linear networks in BAFFLE

We analyze the convergence of BAFFLE in Section 3 using a general technique applicable to any continuously differentiable models corresponding to the loss function $\mathcal{L}(\mathbf{W}; \mathbb{D})$. Since deep networks are the most prevalent models in FL, which has strong linearity, it is simpler to investigate the convergence of deep linear networks [77].

Consider a two-layer deep linear network in a classification task with L categories. We denote the model parameters as $\{\mathbf{W}_1, \mathbf{W}_2\}$, where in the first layer $\mathbf{W}_1 \in \mathbb{R}^{n\times m}$, in the second layer $\mathbf{W}_2 \in \mathbb{R}^{L\times n}$ consists of L vectors related to the L categories as $\{\mathbf{w}_2^l\}_{l=1}^L$ and $\mathbf{w}_2^c \in \mathbb{R}^{1\times n}$. For the input data $\mathbf{X} \in \mathbb{R}^{m\times 1}$ with label y, we train the deep linear network by maximizing the classification score on the y-th class. Since there is no non-linear activation in deep linear networks, the forward inference can be represented as $h = \mathbf{w}_2^y\mathbf{W}_1\mathbf{X}$, and the loss is $-h$. It is easy to show that $\frac{\partial h}{\partial \mathbf{w}_2^y} = (\mathbf{W}_1\mathbf{X})^\top$ and $\frac{\partial h}{\partial \mathbf{W}_1} = (\mathbf{X}\mathbf{w}_2^y)^\top$. We sample $\boldsymbol{\delta}_1, \boldsymbol{\delta}_2$ from noise generator $\mathcal{N}(0, \sigma^2\mathbf{I})$, where $\boldsymbol{\delta}_1 \in \mathbb{R}^{n\times m}$ and $\boldsymbol{\delta}_2 \in \mathbb{R}^{1\times n}$. Let $h(\boldsymbol{\delta}_1, \boldsymbol{\delta}_2) := (\mathbf{w}_2^y + \boldsymbol{\delta}_2)(\mathbf{W}_1 + \boldsymbol{\delta}_1)\mathbf{X}$, we discover that the BAFFLE estimation in

Eq. (6) follows the same pattern for both forward (2) and central schemes (3):

$$\begin{aligned}
&\Delta_{\text{for}} h(\boldsymbol{\delta}_1, \boldsymbol{\delta}_2) := h(\boldsymbol{\delta_1}, \boldsymbol{\delta}_2) - h(\mathbf{0}, \mathbf{0});\\
&\Delta_{\text{ctr}} h(\boldsymbol{\delta}_1, \boldsymbol{\delta}_2) := h(\boldsymbol{\delta}_1, \boldsymbol{\delta}_2) - h(-\boldsymbol{\delta}_1, -\boldsymbol{\delta}_2);\\
&\frac{\Delta_{\text{for}} h(\boldsymbol{\delta}_1, \boldsymbol{\delta}_2)}{\sigma^2} = \frac{\Delta_{\text{ctr}} h(\boldsymbol{\delta}_1, \boldsymbol{\delta}_2)}{2\sigma^2} = \frac{1}{\sigma^2}(\mathbf{w}_2^c \boldsymbol{\delta}_1 \mathbf{X} + \boldsymbol{\delta}_2 \mathbf{W}_1 \mathbf{X}).
\end{aligned} \tag{13}$$

This equivalent form in deep linear networks illustrates that the residual benefit from the central scheme is reduced by the linearity, hence the performance of the two finite difference schemes described above is same in deep linear networks. We refer to this characteristic as FD scheme independence. We also find the property of σ independence, that is, the choice of σ does not effect the results of finite difference, due to the fact that $\frac{\boldsymbol{\delta}_1}{\sigma}$ and $\frac{\boldsymbol{\delta}_2}{\sigma}$ follow the standard normal distribution.

Based on the findings from Eq. (13), we propose the following useful guideline that improves accuracy under the same computation cost: *Using twice forward difference (twice-FD) scheme rather than central scheme.* Combining the forward scheme Eq. (2) and central scheme Eq. (3), we find that the central scheme produces smaller residuals than the forward scheme by executing twice as many forward inferences, i.e. $\mathbf{W} \pm \boldsymbol{\delta}$. With the same forward inference times (e.g., $2K$), one practical difficulty is to identify which scheme performs better. We find that the forward scheme performs better in all experiments, in part because the linearity reduces the benefit from second-order residuals, as demonstrated by Eq. (13).

D Robustness to inference attacks

To explore the information leakage from outputs $\Delta\mathcal{L}$, we design heuristic experiments. Regular attacks such as membership inference attacks and model inversion attacks cannot directly target BAFFLE since they must repeatedly do model inference and get confidence values or classification scores. To analyze the possibility of information leaking, we employ the concept of differential privacy [1] and compare the BAFFLE's outputs from private data to random noise. If we cannot discriminate between private data and random noise merely from the BAFFLE's outputs, we can assert that the outputs do not contain private information. In details, we utilize the validation dataset as the private data and generate random input pairs from Gaussian and Laplacian noise as $(\tilde{\mathbf{X}}, \tilde{y})$. Then we apply BAFFLE to both private data and random noise and compare the distributions of their respective outputs $\Delta\mathcal{L}$. As shown in Figure 5, it is difficult to distinguish the BAFFLE's outputs between private data and random noise, showing that it is difficult for attackers to acquire meaningful information rather than random noise from the BAFFLE's outputs.