A. Method of Inferring Data Representations

As discussed in Section 3.1, if we can find several rows in the local update $\boldsymbol{W}$ that is from $Grad(\mathbb{B}_i)$, which is possible because of the low entanglement of $Grad(\mathbb{B}_i)$ across i in FL, then we are able to infer this device's training data representation of class i in this layer. As $\frac{\partial l^j}{\partial \boldsymbol{b}^j}$ and $(\boldsymbol{r}^j)^{\mathrm{T}}$ are both similar across j in one batch $\mathbb{B}_i$, $Grad(\mathbb{B}_i)$ can be approximated as equation 14,

$$\begin{aligned} Grad(\mathbb{B}_i) &= \frac{1}{|\mathbb{B}_i|}\sum_{j\in\mathbb{B}_i}\frac{\partial l^j}{\partial \boldsymbol{b}^j}\boldsymbol{r}^{j\,\mathrm{T}} \\ &\approx \left(\frac{1}{|\mathbb{B}_i|}\sum_{j\in\mathbb{B}_i}\frac{\partial l^j}{\partial \boldsymbol{b}^j}\right)\left(\frac{1}{|\mathbb{B}_i|}\sum_{j\in\mathbb{B}_i}\boldsymbol{r}^{j\,\mathrm{T}}\right) \\ &= \overline{\frac{\partial l}{\partial \boldsymbol{b}}}_{\mathbb{B}_i}\overline{\boldsymbol{r}}^{\mathrm{T}}_{\mathbb{B}_i}, \end{aligned} \tag{14}$$

where $\overline{\frac{\partial l}{\partial \boldsymbol{b}}}_{\mathbb{B}_i}$ and $\overline{\boldsymbol{r}}^{\mathrm{T}}_{\mathbb{B}_i}$ denote the average of $\frac{\partial l^j}{\partial \boldsymbol{b}^j}$ and $(\boldsymbol{r}^j)^{\mathrm{T}}$ for $j \in \mathbb{B}_i$, and $\overline{\boldsymbol{r}}^{\mathrm{T}}_{\mathbb{B}_i}$ is the data representation corresponding to this device's training data of class i in this layer. If we want to infer $\overline{\boldsymbol{r}}^{\mathrm{T}}_{\mathbb{B}_i}$ from this layer's local parameter update, we need to seek out the unique elements in $\overline{\frac{\partial l}{\partial \boldsymbol{b}}}_{\mathbb{B}_i}$. Here, unique elements are the elements in $\overline{\frac{\partial l}{\partial \boldsymbol{b}}}_{\mathbb{B}_i}$ that are not, or less entangled with other $\overline{\frac{\partial l}{\partial \boldsymbol{b}}}_{\mathbb{B}_{j,j\neq i}}$ after summation in equation 2 is executed.

A.1. Inferring features in the last layer

Let us consider the last layer of a classification model with cross-entropy loss over a sample. Suppose $\boldsymbol{r}$ is the data representation of the second-to-layer layer, we have

$$\begin{aligned} \boldsymbol{b} &= \boldsymbol{W}\boldsymbol{r} \\ \boldsymbol{y} &= softmax(\boldsymbol{b}) \\ l &= -\log \boldsymbol{y}_c, \end{aligned} \tag{15}$$

where l is the loss defined on a sample and c is the sample's ground-truth label. $\boldsymbol{y} = [y_1, y_2, \ldots, y_C]$ denotes the output of the $softmax$. Then $\frac{\partial l}{\partial \boldsymbol{b}}$ in this layer is

$$\frac{\partial l}{\partial \boldsymbol{b}}_i = \begin{cases} y_i - 1, & i = c \\ y_i, & i \neq c \end{cases} \tag{16}$$

As $y_1, y_2, \ldots y_C$ are probabilities, we have $y_i \in (0, 1)$ and $\sum_i y_i = 1$. Hence, $\frac{\partial l}{\partial \boldsymbol{b}}$ has only one negative element on index c and the absolute value of $\frac{\partial l_c}{\partial \boldsymbol{b}}$ is equal to the sum of other elements' absolute values. Therefore, for the last layer, the unique element in $\overline{\frac{\partial l}{\partial \boldsymbol{b}}}_{\mathbb{B}_i}$ is the "peak" element with index i, and this "peak" element contributes to the larger $||\nabla \boldsymbol{W}_i||_2$, where $\nabla \boldsymbol{W}_i$ denotes the i^{th} row of $\nabla \boldsymbol{W}$.

When the malicious server receives one local model updates, it computes $\{||\nabla \boldsymbol{W}_1||_2, ||\nabla \boldsymbol{W}_2||_2, \ldots, ||\nabla \boldsymbol{W}_C||_2\}$ and picks out the ones that are significantly larger. Then the server successfully infer data classes on this device because these selected rows' indexes corresponds to this device's training data classes. For one training class i, $\overline{\boldsymbol{r}}^{\mathrm{T}}_{\mathbb{B}_i}$ in this layer can just be approximated by $\gamma \nabla \boldsymbol{W}_i$, where γ is a scale influences by the local training steps. The algorithm of inferring data representations in the last layer is shown in Algorithm 3.

A.2. Inferring features in previous layers

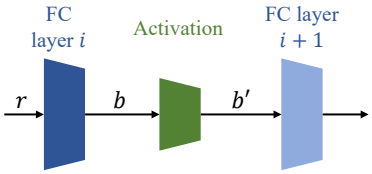

Figure 8. Inference process.

Generally, we need to seek out the unique elements in $\overline{\frac{\partial l}{\partial \boldsymbol{b}}}_{\mathbb{B}_i}$ to infer $\overline{\boldsymbol{r}}^{\mathrm{T}}_{\mathbb{B}_i}$ in this layer. Let us assume we have inferred the data representation of $\mathbb{B}_i$ in the layer after, which is denoted as $\overline{\boldsymbol{b}'}_{\mathbb{B}_i}$ shown in figure 8. Specifically, $\overline{\boldsymbol{b}'}_{\mathbb{B}_i}$ is the result of activation function with input as $\overline{\boldsymbol{b}}_{\mathbb{B}_i}$. If we can infer $\overline{\boldsymbol{r}}^{\mathrm{T}}_{\mathbb{B}_i}$ based on the access of $\overline{\boldsymbol{b}'}_{\mathbb{B}_i}$, plus the inferred last layer's data representation of $\mathbb{B}_i$, then we can infer $\mathbb{B}_i$'s data representations of every linear layer in a backpropagation fashion.

Even though $\overline{\boldsymbol{b}'}_{\mathbb{B}_i}$ is a nonlinear transformation of $\overline{\boldsymbol{b}}_{\mathbb{B}_i}$, they share the similar structure and sparsity due to the consistency of most activation functions. Hence we can apply $\overline{\frac{\partial l}{\partial \boldsymbol{b}'}}_{\mathbb{B}_i}$ to approximate $\overline{\frac{\partial l}{\partial \boldsymbol{b}}}_{\mathbb{B}_i}$ for seeking the unique elements in $\overline{\frac{\partial l}{\partial \boldsymbol{b}}}_{\mathbb{B}_i}$. Theoretically, $\overline{\frac{\partial l}{\partial \boldsymbol{b}'}}_{\mathbb{B}_i}$ corresponds to the direction of $\nabla \boldsymbol{b}'_{\mathbb{B}_i}$. Because $\boldsymbol{b}'_{\mathbb{B}_i}$ should retain stable structure and sparsity in one local updating round as discussed in Section 3.1, $\nabla \boldsymbol{b}'_{\mathbb{B}_i}$ should mostly appear on the elements with larger magnitude. Therefore, the unique elements in $\overline{\frac{\partial l}{\partial \boldsymbol{b}'}}_{\mathbb{B}_i}$ should have the same indexes with the elements with larger magnitude in $\boldsymbol{b}'_{\mathbb{B}_i}$. Since we have access to $\boldsymbol{b}'_{\mathbb{B}_i}$, we can find M most unique elements in $\overline{\frac{\partial l}{\partial \boldsymbol{b}'}}_{\mathbb{B}_i}$ by listing the M elements in $\boldsymbol{b}'_{\mathbb{B}_i}$ with the largest magnitude. Then we can infer $\overline{\boldsymbol{r}}^{\mathrm{T}}_{\mathbb{B}_i}$ easily by fetching and averaging the rows of this layer's weight updates according to the M unique elements indexes.

Following the above algorithm, the malicious server can fetch the training data representation in a fully connected layer for each data class on one device based on the data representation in the layer after. Plus the inference of all classes' training data representations in the last layer, the server is able to infer one device's training data representations for each class it owns in every fully connected layers in a back propagation way. The inferring process is shown in Algorithm 4.

Algorithm 3 Data representation inference in the last layer.

Input: Local weight updates in the last layer $\nabla \boldsymbol{W}$.
Output: Local training class set $\mathbb{S}$; Linearly scale training data representations $\{\hat{\boldsymbol{r}}^{\mathrm{T}}_{\mathbb{B}_i}, i \in \mathbb{S}\}$ in this layer.

1: Compute $||\nabla \boldsymbol{W}_1||_2, ||\nabla \boldsymbol{W}_2||_2, \ldots, ||\nabla \boldsymbol{W}_C||_2$;
2: Pick up peaks of $\{||\nabla \boldsymbol{W}_i||_2\}$ and collect their indexes as $\mathbb{S}$;
3: **return** $\mathbb{S}$, $\{\nabla \boldsymbol{W}_i, i \in \mathbb{S}\}$;

Algorithm 4 Data representation inference in previous fully connected layers.

Input: Local weight updates in this layer $\nabla \boldsymbol{W}$; Data representation $\overline{\boldsymbol{b}'}_{\mathbb{B}_i}$ for $\mathbb{B}_i$ in the following layer; $M \in \mathbb{N}+$.
Output: Linearly scale training data representations $\hat{\boldsymbol{r}}^{\mathrm{T}}_{\mathbb{B}_i}$ in this layer.

1: Select M elements in $\overline{\boldsymbol{b}'}_{\mathbb{B}_i}$ with the largest magnitudes and collect their indexes as $\mathbb{M}$;
2: $\hat{\boldsymbol{r}}^{\mathrm{T}}_{\mathbb{B}_i} \leftarrow \sum_{k \in \mathbb{M}} \nabla \boldsymbol{W}_k$;
3: **return** $\hat{\boldsymbol{r}}^{\mathrm{T}}_{\mathbb{B}_i}$;

B. Experiment Setup

Model for experiments in Section 3.2. For the inferring class-wise data representation experiment, we use the base model with 2 convolutional layers and 3 fully connected layers. The detailed architecture is listed as *Conv3-6→Maxpool→Conv6-16→Maxpool→FC–120→FC–84→FC–10*. We set kernel size as 5 and 2 for all convolutional layers and max pooling layers respectively.

Settings for experiments in Section 3.3. For experiments unveiling representation leakage in Section 3.3, we build a model with one convolutional layer and one fully connected layer. The detailed architecture is listed as *Conv3-12→FC–10*. We set kernel size of the convolutional layer as 5. For attacks, we apply the *L-BFGS* optimizer and conduct 300 iterations of optimization to reconstruct the raw data.

Models for two attacks in Section 6 We use *LeNet* for both the *DLG* attack and *ConvNet* for *GS* attack. The architectures are shown in Tab. 3.

Table 3. Model architectures for *DLG* attack and *GS* attack.

DLG	***GS***
5× 5 Conv 3-12	5× 5 Conv 3-32
5× 5 Conv 12-12	5× 5 Conv 32-64
5× 5 Conv 12-12	5× 5 Conv 64-64
5× 5 Conv 12-12	5× 5 Conv 64-128
FC–10	5× 5 Conv 128-128
	5× 5 Conv 128-128
	3× 3 Maxpool
	5× 5 Conv 128-128
	5× 5 Conv 128-128
	5× 5 Conv 128-128
	3× 3 Maxpool
	FC–10

C. Proof of Theorem 1

Proposition 1. *Let* $||\cdot||_p$ *be a sub-multiplicative norm.* $||AB||_p \leq ||A||_p ||B||_p$.

With Assumption 1 and Lemma 1, the distance between X and X' is:

$$\begin{aligned} ||X - X'||_p &= ||f^{-1}(r) - f^{-1}(r')||_p \\ &= ||\nabla_r f^{-1} \cdot (r - r')||_p \\ &= ||(\nabla_X f)^{-1} \cdot (r - r')||_p \end{aligned} \tag{17}$$

Based on Proposition 1, we have $||C^{-1}D||_p \geq ||D||_p / ||C||_p$. Then, $||X - X'||_p$ is lower bounded as

$$||X - X'||_p \geq \frac{||r - r'||_p}{||\nabla_X f||_p}. \tag{18}$$

D. Proof of Theorem 2

Overview: Our proof is mainly inspired by [15]. Specifically, our proof has two key parts. First, we derive the bounds similar to those in Assumptions 4 and 5, after applying our defense scheme. Second, we adapt **Theorem 2** on convergence guarantee in [15] using our new bounds.

Bounding the expected distance between the perturbed gradients with our defense and raw gradients using Assumption 6. In FedAvg, in the t-th round, we denote the input representation, parameters, and output of the single s-th layer in the k-th device as r_t^k, $\boldsymbol{w}_{st}^k$, and b_t^k, respectively. Via applying our defense scheme $\mathcal{T}(\cdot)$, the input representation is perturbed as ${r'}_t^k$. Then, the expected distance between the perturbed gradients and raw gradients in the s-th layer is bounded by:

$$\mathbb{E}||\nabla F'_k(\boldsymbol{w}_{st}^k, \xi_t^k) - \nabla F_k(\boldsymbol{w}_{st}^k, \xi_t^k)||_2 \tag{19}$$

$$= \mathbb{E}||\nabla_{b_t^k} F_k(\boldsymbol{w}_{st}^k, \xi_t^k) \cdot ({r'}_t^k - r_t^k)^T||_2 \tag{20}$$

$$\leq \mathbb{E}||\nabla_{b_t^k} F_k(\boldsymbol{w}_{st}^k, \xi_t^k)||_2 \cdot ||({r'}_t^k - r_t^k)||_2 \tag{21}$$

$$\leq \Lambda_s \cdot \epsilon, \tag{22}$$

where in Equ. (22) we use the the constraint in Equ. (8) by setting $q = 2$ and Assumption 6.

New bounds for Assumption 4 with our defense. Note that our defense scheme is only applied to the s-th layer. Then, the distance between the perturbed gradients $\nabla F'_k(\boldsymbol{W}_t^k, \xi_t^k)$ and the raw gradients $\nabla F_k(\boldsymbol{W}_t^k, \xi_t^k)$ of the whole model is the same as that of the s-th layer. Thus,

$$\mathbb{E}||\nabla F'_k(\boldsymbol{W}_t^k, \xi_t^k) - \nabla F_k(\boldsymbol{W}_t^k, \xi_t^k)||_2 \tag{23}$$

$$= \mathbb{E}||\nabla F'_k(\boldsymbol{w}_{st}^k, \xi_t^k) - \nabla F_k(\boldsymbol{w}_{st}^k, \xi_t^k)||_2 \tag{24}$$

$$\leq \Lambda_s \cdot \epsilon. \tag{25}$$

Next, we use the norm triangle inequality to bound he variance of stochastic gradients in each device, and we have

$$\begin{aligned}
&\mathbb{E}||\nabla F_k'(\boldsymbol{W}_t^k, \xi_t^k) - \nabla F_k(\boldsymbol{W}_t^k)||^2 && (26)\\
\leq &\mathbb{E}||\nabla F_k'(\boldsymbol{W}_t^k, \xi_t^k) - \nabla F_k(\boldsymbol{W}_t^k, \xi_t^k)||^2 && (27)\\
&+ \mathbb{E}||\nabla F_k(\boldsymbol{W}_t^k, \xi_t^k) - \nabla F_k(\boldsymbol{W}_t^k)||^2 && (28)\\
\leq &\Lambda_s \cdot \epsilon + \sigma_k^2, && (29)
\end{aligned}$$

where we use Assumption 4 and Equ. (25) in Equ. (29).

New bounds for Assumption 5 with our defense. The expected squared norm of stochastic gradients $\nabla F_k'(\boldsymbol{W}_t^k, \xi_t^k)$ with our defense is as follows:

$$\begin{aligned}
&\mathbb{E}||\nabla F_k'(\boldsymbol{W}_t^k, \xi_t^k)||^2 && (30)\\
\leq &\mathbb{E}||\nabla F_k'(\boldsymbol{W}_t^k, \xi_t^k) - \nabla F_k(\boldsymbol{W}_t^k, \xi_t^k)||^2 && (31)\\
&+ \mathbb{E}||\nabla F_k(\boldsymbol{W}_t^k, \xi_t^k)||^2 && (32)\\
\leq &\Lambda_s \cdot \epsilon + G^2, && (33)
\end{aligned}$$

where we use Assumption 5 and and Equ. (25) in Equ. (33).

Convergence guarantee for FedAvg with our defense. We define F^* and F_k^* as the minimum value of F and F_k and let $\Gamma = F^* - \sum_{k=1}^{N} p_k F_k^*$. We assume each device has I local updates and the total number of iterations is T. Let Assumptions 2 to 6 hold and $L, \mu, \sigma_k, G, \Lambda_s$ be defined therein. Choose $\kappa = \frac{L}{\mu}$, $\gamma = \max\{8\kappa, I\}$, the learning rate $\eta_t = \frac{2}{\mu(\gamma+t)}$. By applying our new bounds and **Theorem 2** in [15], FedAvg using our defense has the following convergence guarantee:

$$\mathbb{E}[F(\boldsymbol{W}_T)] - F^* \leq \frac{2\kappa}{\gamma + T}\left(\frac{Q + C}{\mu} + \frac{\mu\gamma}{2}\mathbb{E}||\boldsymbol{W}_0 - \boldsymbol{W}^*||^2\right), \tag{34}$$

where

$$Q = \sum_{k=1}^{N} p_k^2(\Lambda_s \cdot \epsilon + \sigma_k^2) + 6L\Gamma + 8(I-1)^2(\Lambda_s \cdot \epsilon + G^2)$$

$$C = \frac{4}{K} I^2 (\Lambda_s \cdot \epsilon + G^2).$$