Supplementary Material

A List of Symbols

$\mathbf{\Theta}, \mathbf{\Theta}_1, \mathbf{\Theta}_c, \mathbf{b}_1$	Model parameters, parameters of the first FC layer in the model, parameters before the first FC layer in the model, bias of the first FC layer in the model
$\mathbf{H}/\widetilde{\mathbf{H}}, \hat{\mathbf{H}}$	Inputs to the first FC layer by $\mathbf{x}/\hat{\mathbf{x}}$, estimated $\mathbf{H}$ through CAFE step II
$\mathcal{D}, \hat{\mathcal{D}}$	Real, fake dataset
$\mathcal{L}(\cdot)$	Loss function
$\mathcal{M}, M, m$	Set, number, index of local clients
$\nabla_{\Xi}\mathcal{L}(\mathbf{\Theta}, \mathcal{D})$	Gradients of loss function w.r.t. Ξ. Ξ represents $\mathbf{\Theta}, \mathbf{\Theta}_c, \mathbf{\Theta}_1, \mathbf{b}_1, \mathbf{U}$
$\mathbf{s}(\mathbf{s}^t), \mathcal{S}$	Batch index permutation (selected in the tth iteration), batch index permutation sets
$\mathbf{U}$	Outputs of the first FC layer before the activation function
$\mathbf{V}$	Estimated $\nabla_{\mathbf{U}}\mathcal{L}(\mathbf{\Theta}, \mathcal{D})$ through CAFE step I
$\mathbf{x}/\hat{\mathbf{x}}, \mathcal{X}/\hat{\mathcal{X}}$	Real/fake training data (images), real/fake training dataset
d_1, d_2	Inputs, outputs feature dimension of the first FC layer
$h(\cdot)$	Forward function before the first FC layer
K	Batch size
N, n	Number, index of data points
$y, \hat{y}$	Real, fake training labels

B CAFE vs DLG

As in [32], assuming $K = N = 3$, (3) can be rewritten as

$$\hat{\mathcal{D}}^* = \arg\min_{\hat{\mathcal{D}}} \left\| \frac{1}{3}\sum_{n=1}^{3} \nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta}, \mathbf{x}_n, y_n) - \frac{1}{3}\sum_{n=1}^{3} \nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta}, \hat{\mathbf{x}}_n, \hat{y}_n) \right\|^2. \tag{10}$$

We assume that there is a ground-truth solution for (10) denoted as

$$\hat{\mathcal{D}}_1^* = \{\{\mathbf{x}_1, y_1\}; \{\mathbf{x}_2, y_2\}; \{\mathbf{x}_3, y_3\}\}. \tag{11}$$

However, besides the ground-truth solution, there might be other undesired solutions, such as

$$\hat{\mathcal{D}}_2^* = \{\{\hat{\mathbf{x}}_1^*, \hat{y_1}^*\}; \{\hat{\mathbf{x}}_2^*, \hat{y_2}^*\}; \{\mathbf{x}_3, y_3\}\} \tag{12}$$

whose gradients satisfy

$$\begin{aligned} \sum_{n=1}^{2} \nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta}, \mathbf{x}_n, y_n) &= \sum_{n=1}^{2} \nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta}, \hat{\mathbf{x}}_n^*, \hat{y_n}^*) \\ \nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta}, \mathbf{x}_n, y_n) &\neq \nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta}, \hat{\mathbf{x}}_n^*, \hat{y_n}^*). \end{aligned} \tag{13}$$

Although the solutions (11) and (12) have the same objective value in (10), the solution (12) is not the ground-truth solution for data recovery, which needs to be eliminated by introducing more regularization or constraints. When the number N increases, the number of undesired solutions increases. It is hard to find the ground-truth solution by purely optimizing the objective function (10).

However, in CAFE, the number of objective functions can be as many as $\binom{N}{K}$. As the case above, suppose $K = 2$. Then we can list all the objective functions as

$$\begin{cases} \hat{\mathcal{D}}_0^* &= \arg\min_{\hat{\mathcal{D}}_0} \left\| \frac{1}{2}\sum_{n=1}^{2} \nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta}, \mathbf{x}_n, y_n) - \frac{1}{2}\sum_{n=1}^{2} \nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta}, \hat{\mathbf{x}}_n, \hat{y}_n) \right\|^2 \\ \hat{\mathcal{D}}_1^* &= \arg\min_{\hat{\mathcal{D}}_1} \left\| \frac{1}{2}\sum_{n=2}^{3} \nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta}, \mathbf{x}_n, y_n) - \frac{1}{2}\sum_{n=2}^{3} \nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta}, \hat{\mathbf{x}}_n, \hat{y}_n) \right\|^2 \\ \hat{\mathcal{D}}_2^* &= \arg\min_{\hat{\mathcal{D}}_2} \left\| \frac{1}{2}\sum_{n=1,n\neq 2}^{3} \nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta}, \mathbf{x}_n, y_n) - \frac{1}{2}\sum_{n=1,n\neq 2}^{3} \nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta}, \hat{\mathbf{x}}_n, \hat{y}_n) \right\|^2 \end{cases} \tag{14}$$

where $\hat{\mathcal{D}}^0 = \{\{\hat{\mathbf{x}}_1, \hat{y_1}\}; \{\hat{\mathbf{x}}_2, \hat{y_2}\}\}, \hat{\mathcal{D}}^1 = \{\{\hat{\mathbf{x}}_2, \hat{y_2}\}; \{\hat{\mathbf{x}}_3, \hat{y_3}\}\}, \hat{\mathcal{D}}^2 = \{\{\hat{\mathbf{x}}_1, \hat{y_1}\}; \{\hat{\mathbf{x}}_3, \hat{y_3}\}\}$. Comparing with (10), (14) has more constraint functions which restrict $\hat{\mathcal{D}}$ and dramatically reduces the number of undesired solutions. Solution (12) thus can be eliminated by the second and the third equations in (14). It suggests that CAFE helps the fake data converge to the optimal solution.

C Proof of Theorem 1

The second derivative of $\mathcal{F}_1(\mathbf{V})$ w.r.t $\mathbf{V}$ are denoted by

$$\nabla_{v_{p,q};v_{r,s}}\mathcal{F}_1(\mathbf{V}) = \frac{\partial \nabla_{v_{p,q}}\mathcal{F}_1(\mathbf{V})}{\partial v_{r,s}} = \begin{cases} \delta(p,r) & q = s \\ 0 & q \neq s \end{cases} \tag{15}$$

where $v_{p,q}$ is the entry at the pth row and qth column of $\mathbf{V}$ and $\delta(p,r)$ is defined as

$$\delta(p,r) = 2\mathbb{E}_{\mathbf{s}_i \sim \textbf{Unif}(\mathcal{S})}\Big[\mathbf{s}_i[p]\mathbf{s}_i[r]\Big]. \tag{16}$$

The Hessian matrix of the $\mathcal{F}_1(\mathbf{V})$ can be denoted by

$$\nabla^2\mathcal{F}_1(\text{vec}(\mathbf{V})) = \begin{bmatrix} \mathcal{H}(1,1) & \mathcal{H}(1,2) & \dots & \mathcal{H}(1,s) & \dots & \mathcal{H}(1,d_2) \\ \mathcal{H}(2,1) & \mathcal{H}(2,2) & \dots & \mathcal{H}(2,s) & \dots & \mathcal{H}(2,d_2) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathcal{H}(q,1) & \mathcal{H}(q,2) & \dots & \mathcal{H}(q,s) & \dots & \mathcal{H}(q,d_2) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathcal{H}(d_2,1) & \mathcal{H}(d_2,2) & \dots & \mathcal{H}(d_2,s) & \dots & \mathcal{H}(d_2,d_2) \end{bmatrix}_{(N\times d_2)\times(N\times d_2)} \tag{17}$$

where $\text{vec}(\mathbf{V}) \in \mathbb{R}^{(N\times d_2)}$ vectorizes $\mathbf{V}$.

When $q \neq s$, we have $\mathcal{H}(q,s) = \mathbf{0}$. When $q = s$

$$\mathcal{H}(q,s) = \begin{bmatrix} \delta(1,1) & \delta(1,2) & \dots & \delta(1,r) & \dots & \delta(1,N) \\ \delta(2,1) & \delta(2,2) & \dots & \delta(2,r) & \dots & \delta(2,N) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \delta(p,1) & \delta(p,2) & \dots & \delta(p,r) & \dots & \delta(p,N) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \delta(N,1) & \delta(N,2) & \dots & \delta(N,r) & \dots & \delta(N,N) \end{bmatrix}_{N\times N} \tag{18}$$

It is obvious that $\forall q_1 \neq q_2, \mathcal{H}(q_1,q_1) = \mathcal{H}(q_2,q_2)$.
Therefore, we have

$$\nabla^2\mathcal{F}_1(\text{vec}(\mathbf{V})) = \begin{bmatrix} \mathcal{H}(1,1) & \mathbf{0} & \dots & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathcal{H}(1,1) & \dots & \mathbf{0} & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \dots & \mathcal{H}(1,1) & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \dots & \mathcal{H}(1,1) \end{bmatrix}_{(N\times d_2)\times(N\times d_2)}. \tag{19}$$

For any vector $\mathbf{p} = [\mathbf{p}_1^\top, \dots, \mathbf{p}_q^\top, \dots, \mathbf{p}_{d_2}^\top]^\top \neq \mathbf{0} \in \mathbb{R}^{(N\times d_2)}$, where $\mathbf{p}_q \in \mathbb{R}^N$, we have

$$\begin{aligned} \mathbf{p}^\top\nabla^2\mathcal{F}_1(\text{vec}(\mathbf{V}))\mathbf{p} &= \sum_{q=1}^{d_2}\mathbf{p}_q^\top\mathcal{H}(q,q)\mathbf{p}_q \\ &= \sum_{q=1}^{d_2}\mathbf{p}_q^\top\mathcal{H}(1,1)\mathbf{p}_q. \end{aligned} \tag{20}$$

If $\mathcal{H}(1,1)$ is positive definite, then we have $\nabla^2\mathcal{F}_1(\text{vec}(\mathbf{V}))$ is positive definite. Since $\forall \mathbf{s}_i, p, \mathbf{s}_i[p] \in \{0,1\}$, when $p = r$, we have

$$\delta(p,r) = \delta(p,p) = 2\mathbb{E}_{\mathbf{s}_i \sim \textbf{Unif}(\mathcal{S})}\Big[\mathbf{s}_i[p]\Big] = \frac{2K}{N}; \tag{21}$$

when $p \neq r$, we have

$$\delta(p,r) = 2\mathbb{E}_{\mathbf{s}_i \sim \textbf{Unif}(\mathcal{S})}\Big[\mathbf{s}_i[p]\mathbf{s}_i[r]\Big] = 2\frac{\binom{K}{2}}{\binom{N}{2}} = \frac{2K(K-1)}{N(N-1)} \tag{22}$$

As the results, we have

$$\mathcal{H}(1,1) = 2\begin{bmatrix} \frac{K}{N} & \frac{K(K-1)}{N(N-1)} & \cdots & \frac{K(K-1)}{N(N-1)} & \cdots & \frac{K(K-1)}{N(N-1)} \\ \frac{K(K-1)}{N(N-1)} & \frac{K}{N} & \cdots & \frac{K(K-1)}{N(N-1)} & \cdots & \frac{K(K-1)}{N(N-1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{K(K-1)}{N(N-1)} & \frac{K(K-1)}{N(N-1)} & \cdots & \frac{K}{N} & \cdots & \frac{K(K-1)}{N(N-1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{K(K-1)}{N(N-1)} & \frac{K(K-1)}{N(N-1)} & \cdots & \frac{K(K-1)}{N(N-1)} & \cdots & \frac{K}{N} \end{bmatrix}_{N\times N}. \tag{23}$$

If $K = 1$, we have

$$\mathbb{E}_{\mathbf{s}^t}[\mathcal{H}(1,1)] = 2\begin{bmatrix} \frac{K}{N} & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \frac{K}{N} & \cdots & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \frac{K}{N} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & \cdots & \frac{K}{N} \end{bmatrix}_{N\times N} = \frac{2K}{N} I_{N\times N} \tag{24}$$

where $I_{N\times N}$ is the N dimensional identity matrix. Hence, $\mathcal{H}(1,1)$ is positive definite. If $1 < K < N$, we have

$$\mathcal{H}(1,1) = 2\frac{K(K-1)}{N(N-1)}\begin{bmatrix} \frac{N-1}{K-1} & 1 & \cdots & 1 & \cdots & 1 \\ 1 & \frac{N-1}{K-1} & \cdots & 1 & \cdots & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & 1 & \cdots & \frac{N-1}{K-1} & \cdots & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & 1 & \cdots & 1 & \cdots & \frac{N-1}{K-1} \end{bmatrix}_{N\times N}. \tag{25}$$

The eigenvalues of $\mathcal{H}(1,1)$ in (25) are denoted by

$$\begin{aligned} \lambda_1 = \cdots = \lambda_{N-1} &= \frac{N-1}{K-1} - 1 > 0 \\ \lambda_N &= \frac{N-1}{K-1} + N - 1 > 0 \end{aligned} \tag{26}$$

which implies that $\mathcal{F}_1(\text{vec}(\mathbf{V}))$ is strongly convex.

Notably, when $K = N$, we have

$$\mathcal{H}(1,1) = 2\frac{K(K-1)}{N(N-1)}\boldsymbol{J}_N, \tag{27}$$

where $\boldsymbol{J}_N$ is the $N \times N$ dimensional matrix of ones which is not positive definite.

D Proof of Theorem 2

Similar as the term in (15), the second derivative of $\mathcal{F}_2(\hat{\mathbf{H}})$ w.r.t $\hat{\mathbf{H}}$ can be defined as

$$\nabla_{\hat{h}_{p,q};\hat{h}_{r,s}}\mathcal{F}_2(\hat{\mathbf{H}}) = \frac{\partial \nabla_{\hat{h}_{p,q}}\mathcal{F}_2(\hat{\mathbf{H}})}{\partial \hat{h}_{r,s}} = \begin{cases} \omega(p,r) & q = s \\ 0 & q \neq s \end{cases}. \tag{28}$$

where $\hat{h}_{p,q}$ is the element at the pth row and qth column in $\hat{\mathbf{H}}$ and $\omega(p,r)$ is defined as

$$\begin{aligned} \omega(p,r) &= 2\mathbb{E}_{\mathbf{s}_i\sim\textbf{Unif}(\mathcal{S})}\Big[\sum_{k=1}^{d_2}\mathbf{s}_i[p]\mathbf{s}_i[r]v_{p,k}v_{r,k}\Big] \\ &= 2\mathbb{E}_{\mathbf{s}_i\sim\textbf{Unif}(\mathcal{S})}\Big[\mathbf{s}_i[p]\mathbf{s}_i[r]\Big]\sum_{k=1}^{d_2}v_{p,k}v_{r,k} \\ &= \delta(p,r)\sum_{k=1}^{d_2}v_{p,k}v_{r,k}. \end{aligned} \tag{29}$$

The Hessian matrix of the $\mathcal{F}_2(\hat{\mathbf{H}})$ can be denoted by

$$\nabla^2\mathcal{F}_2(\text{vec}(\hat{\mathbf{H}})) = \begin{bmatrix} \mathcal{G}(1,1) & \mathcal{G}(1,2) & \dots & \mathcal{G}(1,s) & \dots & \mathcal{G}(1,d_1) \\ \mathcal{G}(2,1) & \mathcal{G}(2,2) & \dots & \mathcal{G}(2,s) & \dots & \mathcal{G}(2,d_1) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathcal{G}(q,1) & \mathcal{G}(q,2) & \dots & \mathcal{G}(q,s) & \dots & \mathcal{G}(j,d_1) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathcal{G}(d_1,1) & \mathcal{G}(d_1,2) & \dots & \mathcal{G}(d_1,s) & \dots & \mathcal{G}(d_1,d_1) \end{bmatrix}_{(N\times d_1)\times(N\times d_1)} . \tag{30}$$

When $q \neq s$, we have $\mathcal{G}(q,s) = \mathbf{0}$. When $q = s$

$$\mathcal{G}(q,s) = \begin{bmatrix} \omega(1,1) & \omega(1,2) & \dots & \omega(1,r) & \dots & \omega(1,N) \\ \omega(2,1) & \omega(2,2) & \dots & \omega(2,r) & \dots & \omega(2,N) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \omega(p,1) & \omega(p,2) & \dots & \omega(p,r) & \dots & \omega(p,N) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \omega(N,1) & \omega(N,2) & \dots & \omega(N,r) & \dots & \omega(N,N) \end{bmatrix}_{N\times N} . \tag{31}$$

It is obvious that $\forall q_1 \neq q_2, \mathcal{G}(q_1,q_1) = \mathcal{G}(q_2,q_2)$.
Therefore, we have

$$\nabla^2\mathcal{F}_2(\text{vec}(\hat{\mathbf{H}})) = \begin{bmatrix} \mathcal{G}(1,1) & \mathbf{0} & \dots & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathcal{G}(1,1) & \dots & \mathbf{0} & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \dots & \mathcal{G}(1,1) & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \dots & \mathcal{G}(1,1) \end{bmatrix}_{(N\times d_1)\times(N\times d_1)} \tag{32}$$

for any $\mathbf{p} = [\mathbf{p}_1^\top, \dots, \mathbf{p}_q^\top, \dots, \mathbf{p}_{d_1}^\top]^\top \neq \mathbf{0} \in \mathbb{R}^{(N\times d_1)}$, where $\mathbf{p}_q \in \mathbb{R}^N$, we have

$$\begin{aligned} \mathbf{p}^\top\nabla^2\mathcal{F}_2(\text{vec}(\hat{\mathbf{H}}))\mathbf{p} &= \sum_{q=1}^{d_1} \mathbf{p}_q^\top \mathcal{G}(q,q)\mathbf{p}_q \\ &= \sum_{q=1}^{d_1} \mathbf{p}_q^\top \mathcal{G}(1,1)\mathbf{p}_q. \end{aligned} \tag{33}$$

Therefore, if $\mathcal{G}(1,1)$ is positive definite, $\nabla^2\mathcal{F}_2(\text{vec}(\hat{\mathbf{H}}))$ is positive definite. We can rewrite $\mathcal{G}(1,1)$ as

$$\mathcal{G}(1,1) = \mathcal{H}(1,1) \odot \mathcal{R} \tag{34}$$

where $\odot$ is the Hadamard product and $\mathcal{R}$ is defined as

$$\mathcal{R} = \begin{bmatrix} \sum_{k=1}^{d_2} v_{1,k}v_{1,k} & \sum_{k=1}^{d_2} v_{1,k}v_{2,k} & \dots & \sum_{k=1}^{d_2} v_{1,k}v_{r,k} & \dots & \sum_{k=1}^{d_2} v_{1,k}v_{N,k} \\ \sum_{k=1}^{d_2} v_{2,k}v_{1,k} & \sum_{k=1}^{d_2} v_{2,k}v_{2,k} & \dots & \sum_{k=1}^{d_2} v_{2,k}v_{r,k} & \dots & \sum_{k=1}^{d_2} v_{2,k}v_{N,k} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \sum_{k=1}^{d_2} v_{i,k}v_{1,k} & \sum_{k=1}^{d_2} v_{i,k}v_{2,k} & \dots & \sum_{k=1}^{d_2} v_{i,k}v_{r,k} & \dots & \sum_{k=1}^{d_2} v_{i,k}v_{N,k} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \sum_{k=1}^{d_2} v_{N,k}v_{1,k} & \sum_{k=1}^{d_2} v_{N,k}v_{2,k} & \dots & \sum_{k=1}^{d_2} v_{N,k}v_{r,k} & \dots & \sum_{k=1}^{d_2} v_{N,k}v_{N,k} \end{bmatrix}_{N\times N} . \tag{35}$$

According to Schur Product Theorem, since $\mathcal{H}(1,1)$ has been proved to be positive definite in Appendix C, $\mathcal{G}(1,1)$ is positive definite if $\mathcal{R}$ is positive definite. In addition, since $\mathcal{R} = \mathbf{V}(\mathbf{V})^\top$, when $N < d_2$ and $\text{Rank}(\mathbf{V}) = N$, $\mathcal{R}$ and $\mathcal{G}(1,1)$ are positive definite.

E Theoretical Guarantee on Data Recovery for CAFE

E.1 Performance Guarantee for CAFE step I

We assume the stopping criterion for CAFE step I is denoted by

$$\mathcal{F}_1(\mathbf{V};\mathbf{s}_i) = \left\|\mathbf{V}^\top\mathbf{s}_i - \nabla_{\mathbf{b}_1}\mathcal{L}(\mathbf{\Theta},\mathcal{D}(\mathbf{s}_i))\right\|_2^2 < \phi_1,\ \forall \mathbf{s}_i. \tag{36}$$

Then we have

$$\mathcal{F}_1(\mathbf{V}) = \mathbb{E}_{\mathbf{s}_i\sim\mathbf{Unif}(\mathcal{S})}\mathcal{F}_1(\mathbf{V};\mathbf{s}_i) = \frac{K}{N}\|\mathbf{V}-\mathbf{V}^*\|_F^2 \leq \phi_1, \tag{37}$$

where $\mathbf{V}^*$ is the ground truth.

For a given recovery precision for $\mathbf{V}$ as ϵ_1 denoted by $\|\mathbf{V}-\mathbf{V}^*\|_F^2 := \epsilon_1$. We have

$$\epsilon_1 \leq \frac{N}{K}\phi_1. \tag{38}$$

As the result the recovery of $\mathbf{V}$ is guaranteed.

E.2 Performance Guarantee for CAFE step II

We assume the stopping criterion for CAFE step II as ϕ_2 denoted by

$$\forall i, \mathcal{F}_2(\hat{\mathbf{H}};\mathbf{s}_i) = \left\|\sum_{n=1}^{N}\mathbf{s}_i[n]\hat{\mathbf{h}}_n\mathbf{v}_n^\top - \nabla_{\mathbf{\Theta}_1}\mathcal{L}(\mathbf{\Theta},\mathcal{D}(\mathbf{s}_i))\right\|_F^2 < \phi_2. \tag{39}$$

Then we define

$$\mathbf{\Delta} = \sum_{n=1}^{N}\hat{\mathbf{h}}_n\mathbf{v}_n^\top - \nabla_{\mathbf{\Theta}_1}\mathcal{L}(\mathbf{\Theta},\mathcal{D}) = (\hat{\mathbf{H}})^\top\mathbf{V} - (\hat{\mathbf{H}}^*)^\top\mathbf{V}^*. \tag{40}$$

According to (39), we have

$$\mathcal{F}_2(\hat{\mathbf{H}}) = \mathbb{E}_{\mathbf{s}_i\sim\mathbf{Unif}(\mathcal{S})}\mathcal{F}_2(\hat{\mathbf{H}};\mathbf{s}_i) = \frac{K}{N}\|\mathbf{\Delta}\|_F^2 < \phi_2. \tag{41}$$

We assume that for $\mathbf{V}$ and $\mathbf{V}^*$, $N < d_2$ and $\mathrm{Rank}(\mathbf{V}) = \mathrm{Rank}(\mathbf{V}^*) = N$. Then there exist $\mathbf{V}^{-1}$ and $(\mathbf{V}^*)^{-1}$ such that

$$\mathbf{V}\mathbf{V}^{-1} = I_N, \qquad \mathbf{V}^*(\mathbf{V}^*)^{-1} = I_N. \tag{42}$$

We assume that $\|\nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta},\mathcal{D})\|_F^2$, $\|\mathbf{V}^{-1}\|_F^2$ and $\|(\mathbf{V}^*)^{-1}\|_F^2$ are upper bounded by constants $\lambda_{\mathbf{\Theta}}$, $\lambda_{\mathbf{V}}$ and λ_* respectively. For stopping criterions ϕ_1 and ϕ_2, the recovery precision of $\hat{\mathbf{H}}$ is bounded by

$$\|\hat{\mathbf{H}}-\hat{\mathbf{H}}^*\|_F^2 \leq 2\frac{N}{K}(\lambda_{\mathbf{\Theta}}\lambda_{\mathbf{V}}\lambda_*\phi_1 + \lambda_{\mathbf{V}}\phi_2). \tag{43}$$

Proof: First, we have

$$\begin{aligned}
\|\hat{\mathbf{H}}-\hat{\mathbf{H}}^*\|_F^2 &= \|(\hat{\mathbf{H}})^\top - (\hat{\mathbf{H}}^*)^\top\|_F^2 \\
&= \|(\hat{\mathbf{H}})^\top\mathbf{V}\mathbf{V}^{-1} - (\hat{\mathbf{H}}^*)^\top\mathbf{V}^*(\mathbf{V}^*)^{-1}\|_F^2 \\
&= \|((\nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta},\mathcal{D}) + \mathbf{\Delta})\mathbf{V}^{-1} - (\nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta},\mathcal{D}))(\mathbf{V}^*)^{-1}\|_F^2 \\
&= \|(\nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta},\mathcal{D}))(\mathbf{V}^{-1} - (\mathbf{V}^*)^{-1}) + \mathbf{\Delta}\mathbf{V}^{-1}\|_F^2 \\
&\leq 2\|\nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta},\mathcal{D})\|_F^2\|(\mathbf{V}^{-1} - (\mathbf{V}^*)^{-1})\|_F^2 + 2\|\mathbf{\Delta}\|_F^2\|\mathbf{V}^{-1}\|_F^2
\end{aligned} \tag{44}$$

Since

$$\begin{aligned}
\|\mathbf{V}^{-1} - (\mathbf{V}^*)^{-1}\|_F^2 &= \|\mathbf{V}^{-1}(\mathbf{V}^* - \mathbf{V})(\mathbf{V}^*)^{-1}\|_F^2 \\
&\leq \|\mathbf{V}^{-1}\|_F^2\|(\mathbf{V}^*)^{-1}\|_F^2\|\mathbf{V}^* - \mathbf{V}\|_F^2
\end{aligned} \tag{45}$$

we have

$$\|\hat{\mathbf{H}}-\hat{\mathbf{H}}^*\|_F^2 \leq 2\|\nabla_{\mathbf{\Theta}}\mathcal{L}(\mathbf{\Theta},\mathcal{D})\|_F^2\|\mathbf{V}^{-1}\|_F^2\|(\mathbf{V}^*)^{-1}\|_F^2\|\mathbf{V}^* - \mathbf{V}\|_F^2 + 2\|\mathbf{\Delta}\|_F^2\|\mathbf{V}^{-1}\|_F^2. \tag{46}$$

F Defense Algorithm Based on Fake Gradients

In this section, we list the pseudo-code of our defense strategy in Section 3.4.

Algorithm 5 VFL with fake gradients (in the t-th iteration)

Require: training dataset $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^N$, number of local clients M, model parameters $\mathbf{\Theta}^t$, loss function $\mathcal{L}(\mathcal{D}, \mathbf{\Theta}^t)$, number of fake gradients ν, L_2 distance threshold τ

$\Psi \leftarrow$ construct ν gradients with entries being i.i.d. drawn from $\mathcal{N}(0, \sigma^2)$
For each gradient in Ψ, we sort its elements in descending order
Generate batch indices $\mathbf{s}^t$
for $m = 1, 2, \ldots, M$ **do**
  Worker m takes real batch data
  Worker m exchanges intermediate results to compute local gradients $\nabla_{\mathbf{\Theta}}\mathcal{L}(\mathcal{D}(\mathbf{s}^t), \mathbf{\Theta}^t)$.
  sort-indexes $\boldsymbol{\zeta} \leftarrow \text{argsort}\left|\nabla_{\mathbf{\Theta}}\mathcal{L}(\mathcal{D}(\mathbf{s}^t), \mathbf{\Theta}^t)\right|$ (descending order)
  while $\text{argmin}_{\boldsymbol{\psi}\in\Psi}\left\|\boldsymbol{\psi} - \nabla_{\mathbf{\Theta}}\mathcal{L}(\mathcal{D}(\mathbf{s}^t), \mathbf{\Theta}^t)[\boldsymbol{\zeta}]\right\|_2 > \tau$ **do**
    $\Psi \leftarrow$ construct ν gradients with entries being i.i.d. drawn from $\mathcal{N}(0, \sigma^2)$
    For each gradient in Ψ, we sort its elements in descending order
  end while
  $\boldsymbol{\psi} \leftarrow \text{argmin}_{\boldsymbol{\psi}\in\Psi}\left\|\boldsymbol{\psi} - \nabla_{\mathbf{\Theta}}\mathcal{L}(\mathcal{D}(\mathbf{s}^t), \mathbf{\Theta}^t)[\boldsymbol{\zeta}]\right\|_2$
  initialize fake gradients $\mathbf{g} \leftarrow \mathbf{0}$ {$\mathbf{g}$ has the same dimension as $\nabla_{\mathbf{\Theta}}\mathcal{L}(\mathcal{D}(\mathbf{s}^t), \mathbf{\Theta}^t)$}
  for $i = 1, 2, \ldots, |\boldsymbol{\zeta}|$ **do**
    initialize gradients index $\ell \leftarrow 0$
    for k in $\boldsymbol{\zeta}[i]$ **do**
      $\mathbf{g}[i][k] = \min(\boldsymbol{\psi}[i][\ell], \max(\nabla_{\mathbf{\Theta}}\mathcal{L}(\mathcal{D}(\mathbf{s}^t), \mathbf{\Theta}^t)[i][k], -\boldsymbol{\psi}[i][\ell]))$
      $\ell = \ell + 1$
    end for
  end for
  Upload $\mathbf{g}$ to the server.
end for

G Additional Details on Experiments

In this section, we will provide additional details on the experiments that cannot fit in the main paper.

G.1 Choices of hyper-parameters

Table 9: Choice of hyper-parameters on CAFE
($M = 4$, $K = 40$, batch ratio = 0.05, Nested-loops)

Hyper-parameter / Terms / Method	lr of Step I, II, III	$\alpha, \beta, \gamma, \xi$
CIFAR-10	$5 \times 10^{-3}, 8 \times 10^{-3}, 2 \times 10^{-2}$	$10^{-2}, 10^{-4}, 10^{-3}, 90$
MNIST	$10^{-2}, 10^{-2}, 10^{-2}$	$10^{-2}, 10^{-4}, 10^{-3}, 25$
Linnaeus 5	$5 \times 10^{-3}, 5 \times 10^{-3}, 10^{-2}$	$10^{-2}, 10^{-4}, 10^{-3}, 110$
Yale dataset 32×32	$10^{-2}, 10^{-2}, 10^{-2}$	$10^{-2}, 10^{-4}, 10^{-3}, 32$

We list the choice of hyper-parameters on CAFE ($M = 4$, $K = 40$, Nested-loops) in Table 9. The hyper-parameters of other experiments such as ablation study are adjusted based on these settings.

G.2 Experiments of CAFE PSNR via epoch

In Table 3, we fixed the number T for each dataset and it shows that large batch size indeed helps the CAFE algorithm to approximate $\mathbf{H}$, especially in MNIST. We also conducted an experiment using the same number of epochs on Linnaeus 5 (same setup in Table 3) and reported the results in Table 10. The results suggest that increasing batch size K and number of iterations T both contribute to the

Table 10: Effect of T
(Linnaeus 5, 800 data samples in total)

PSNR \ K / Epoch	10	20	40	80	100
100	12.30	14.76	15.33	11.84	11.79
150	15.83	17.92	16.26	14.28	13.21
200	17.63	19.38	17.20	16.24	14.46
250	21.80	21.49	19.09	18.11	16.14
300	22.92	24.00	21.14	19.83	17.29
350	24.86	25.86	22.62	21.05	18.90

Table 11: Training loss via DP

Training loss \ DP / # of iterations	DP $\epsilon = 10$	DP $\epsilon = 5$	DP $\epsilon = 1$	DP $\epsilon = 0.1$	Fake gradients	True gradients
0	2.78	2.77	2.77	2.77	2.77	2.77
1000	2.69	2.69	2.69	2.69	1.95	1.08
2000	2.85	2.85	2.85	2.85	1.38	0.54
3000	2.85	2.85	2.85	2.85	0.65	0.23
4000	2.92	2.92	2.92	2.92	1.09	0.38
6000	2.69	2.69	2.69	2.69	0.62	0.31
8000	2.69	2.69	2.69	2.69	1.15	0.46

attack performance. When we fix the number of epochs, the attacker with a smaller batch size needs more iterations to recover data, leading to a better performance.

G.3 Comparison with DP-based defense

The results in Table 11 show the training loss of no defense (true gradients), differential privacy (DP) defense, and our defense (fake gradients). For DP, we followed the gradient clipping approach [1] to apply DP to the gradients from workers. In particular, the gradient norm was clipped to 3, as suggested by [1]. As shown in Table 11, the training loss cannot be effectively reduced using DP. This is also consistent with the result in [32] which adds noise to gradients as a candidate defense. However, to avoid information leakage from gradients, the noise magnitude needs to be above a certain threshold which will degrade the accuracy significantly. As the noise magnitude required by DP is even stronger than the one needed for the ad hoc privacy in [32], it is inevitable to lead to a similar conclusion. In our fake gradients defense, all of the gradients will be projected to a set of predefined gradients before being sent to the server, with the purpose of restricting the attacker's knowledge from gradients leakage. Our defense is still deterministic in its essence and therefore does not satisfy the DP. In sum, our experiments demonstrate that the attacker is unable to recover the worker's data and at the same time the training loss can be reduced effectively.

G.4 Experiments on human face dataset

Real data image 1-5

Recovered data image 1-5

Real data image 6-25

Recovered data image 6-25

Figure 8: Visualization of CAFE on Yale 32×32 human face dataset