Supplementary Materials for Breaching FedMD: Image Recovery via Paired-Logits Inversion Attack

Hideaki Takahashi[*] Jingjing Liu[†] Yang Liu[†]

A Deviation of Eq. 5

Proof. We solve Eq. 5 independently for $p^k_{j,\tau}$ and $p^0_{j,\tau}$. The optimal $p^k_{j,\tau}$ is obviously as follows;

$$\hat{p}^k_{\underline{u},\tau} = \begin{cases} 1 & (u = j) \\ 0 & (u \neq j) \end{cases} \tag{S-1}$$

Next, we solve the optimal $p^0_{j,\tau}$ under the constraint of its sum equal to one.

$$\max_{p^0_{j,\tau}} p^0_{\underline{j},\tau} + \alpha H(p^0_{j,\tau}) \tag{S-2}$$

$$s.t. \sum_{u=1}^{J} p^0_{\underline{u},\tau} = 1 \tag{S-3}$$

Substituting Eq. S-3, we can arrange Eq. S-2 as follows;

$$\max_{p^0_{j,\tau}} \quad 1 - \sum_{u=1,u\neq j}^{J} p^0_{\underline{u},\tau} - \alpha\Big(\sum_{u=1,u\neq j}^{J} p^0_{\underline{u},\tau} \log p^0_{\underline{u},\tau}\Big) \tag{S-4}$$

$$- \alpha\Big(1 - \sum_{u=1,u\neq j}^{J} p^0_{\underline{u},\tau}\Big) \log\Big(1 - \sum_{u=1,u\neq j}^{J} p^0_{\underline{u},\tau}\Big) \tag{S-5}$$

, which requires the bellow for all $u \neq j$:

$$-1 + \alpha(\log p^0_{\underline{j},\tau} + 1) - \alpha(\log p^0_{\underline{u},\tau} + 1) = 0 \tag{S-6}$$

Then, we have that

$$\forall u \in \{u : 1 \leq u \leq J, u \neq j\}, \quad \sqrt[\alpha]{e} = \frac{p^0_{\underline{j},\tau}}{p^0_{\underline{u},\tau}} \tag{S-7}$$

Eq. S-3 and S-7 lead to

$$\hat{p}^0_{\underline{u},\tau} = \begin{cases} \frac{\sqrt[\alpha]{e}}{J-1+\sqrt[\alpha]{e}} & (u = j) \\ \frac{1}{J-1+\sqrt[\alpha]{e}} & (u \neq j) \end{cases}$$

□

B Protocols & Architectures

Algorithm S-1 FedMD

Input: Private datasets $\{D_k\}^C_{k=1}$, public dataset D_p, local models $\{f_k\}^C_{k=1}$, global model f_0, number of communications T.

1: Each client trains f_k on D_p
2: Each client trains f_k on D_k
3: **for** $t = 1 \leftarrow T$ **do**
4: Each client sends the set of public logits $\{\boldsymbol{l}^k_i\}$
5: The server computes the global logits:
6: $\boldsymbol{l}_p = \frac{1}{K}\sum^K_{k=1} \boldsymbol{l}^k$
7: Each client receives $\boldsymbol{l}_p$ and trains f_k on $\{D_p, \boldsymbol{l}_p\}$
8: Each client trains f_k on D_k
9: The server trains f_0 on D_p

Algorithm S-2 FedGEMS

Input: Private datasets $\{D_k\}^K_{k=1}$, public dataset D_p, local models $\{f_k\}^K_{k=1}$, global model f_0, number of communications T.

for $t = 1 \leftarrow T$ **do**
The server selectively trains f_0 on $\{D_p, \boldsymbol{l}_p, \boldsymbol{l}_k\}$
The server computes the global logits:
$\boldsymbol{l}^p_i = f_0(W_p; x^p_i)$
Each client trains f_k on $\{D_p, \boldsymbol{l}_p\}$
Each client trains f_k on D_k
Each client sends the set of public logits $\{\boldsymbol{l}^k_i\}$

Alg. S-1, S-2, and S-3 are the pseudo-codes of each protocol, where we additionally train the global model on the public dataset at line 9 in FedMD. Code. 1 and 2 are the implementation of global, local, and inversion models.

[*]The University of Tokyo
takahashi-hideaki567@g.ecc.u-tokyo.ac.jp
[†]Institute for AI Industry Research, Tsinghua University, {jjliu,liuy03}@air.tsinghua.edu.cn

Algorithm S-3 DS-FL

Input: Private datasets $\{D_k\}_{k=1}^{K}$, public dataset D_p, local models $\{f_k\}_{k=1}^{K}$, global model f_0, number of communications T.

1: **for** $t = 1 \leftarrow T$ **do**
2: Each client trains f_k on $\{D_k\}$
3: Each client sends the set of public logits $\{\boldsymbol{l}_i^k\}$
4: The server computes the global logits:
5: $\boldsymbol{l}_p = \text{ERA}(\sum_{k=1}^{K} \frac{\boldsymbol{l}^k}{K})$
6: Each client trains f_k on $\{D_p, \boldsymbol{l}_p\}$
7: The server trains f_0 on $\{D_p, \boldsymbol{l}_p\}$

Code 1. Server and local models

```
nn.Sequential(
    nn.Conv2d(3, 32, kernel_size=(3, 3),
     stride=1, padding=0),
    nn.ReLU(),
    nn.MaxPool2d(kernel_size=(3, 3),
     stride=None, padding=0),
    nn.Flatten(),
    nn.Linear(12800, output_dim))
```

Code 2. Inversion model

```
nn.Sequential(
    nn.ConvTranspose2d(input_dim, 1024,
     (4, 4), stride=(1, 1)),
    nn.BatchNorm2d(1024),
    nn.Tanh(),
    nn.ConvTranspose2d(1024, 512, (4, 4),
     stride=(2, 2), padding=(1, 1)),
    nn.BatchNorm2d(512),
    nn.Tanh(),
    nn.ConvTranspose2d(512, 256, (4, 4),
     stride=(2, 2), padding=(1, 1)),
    nn.BatchNorm2d(256),
    nn.Tanh(),
    nn.ConvTranspose2d(256, 128, (4, 4),
     stride=(2, 2), padding=(1, 1)),
    nn.BatchNorm2d(128),
    nn.Tanh(),
    nn.ConvTranspose2d(128, 3, (4, 4),
     stride=(2, 2), padding=(1, 1)),
    nn.Tanh())
```

C Hyper Parameters

For FedMD, the number of consensuses, revisit, and server-side epochs are 1, and the number of transfer epochs is 5. For FedGEMS, the client-side epoch on both public and private datasets are 2, and the number of server-side epochs is 1. For DS-FL, the number of epochs of local update and distillation is 2, and the number of epochs of server-side distillation is 1. Thus, local models iterate both datasets ten times, and the server iterates the public dataset 5 times in all settings.

We use Adam optimizer with a learning rate of 1e-3 and batch size of 64. The number of clients is 1 or 10. Following the original papers, we set the parameter of FedGEMS ϵ to 0.75 and the temperature of DS-FL to 0.1. Although the original FedMD does not use a server-side model, we train a server-side model on the labeled public dataset. The number of communication is 5 in all schemes.

For PLI, the attacker trains the same architecture (Code. 2 in Appendix B) used in the original TBI as G_θ, with Adam optimizer whose learning rate is 3e-5, weight-decay is 1e-4, and batch size is 8. We experiment with 0.3, 1, 3, and 5 for temperature τ, 0.0, 0.03, 0.1, 0.3, and 1.0 for γ, 5.0 for α, and 0.1 for β. The number of epochs M in each communication is 3. For CycleGAN and DeblurGAN-v2, We set a learning rate of 1e-4 and 2e-4 for each, with a batch size of 1 and 100 epochs.

For comparison, we attack the victim with TBI with the same model architecture, optimizer, and data augmentation. As in the original paper, TBI trains a single inversion model with all available logits on the public dataset. We also apply gradient inversion attacks to FedAVG, the standard scheme of FL, as the baseline. Same as other FedMD-like schemes, the number of communication is 3, and the epoch of local training is 2, but we do not use the public dataset in FedAVG (See Appendix F for details).

D Additional Results

Impact of γ and feature space gap As discussed in Sec. 3.3, the better the quality of prior images is, the higher effective γ is. Tab. S-1 and Tab. S-2 show the results of attack accuracy and SSIM with different γ, which are visualized in Fig. 7. The recovered images with different γ can be found in Fig. S-1. Tab. S-3 also reports the SSIM between the GAN-based prior images based on the labeled public dataset and the private images. It is natural to assume that this SSIM correlates to the feature space gap since the smaller feature space gap between the sensitive and insensitive data should improve the quality of prior data. Thus, these tables validate the proportional relationship between the feature space gap and the optimal γ for attack accuracy.

Impact of τ As stated in Sec. 3.4, τ controls the trade-off between the quality and the accuracy. Tab. S-4 and Tab. S-5 report the numerical values of attack accuracy and SSIM with different τ, which are summarized in Fig.9. Fig. S-2 also shows the reconstruction error and the intermediate

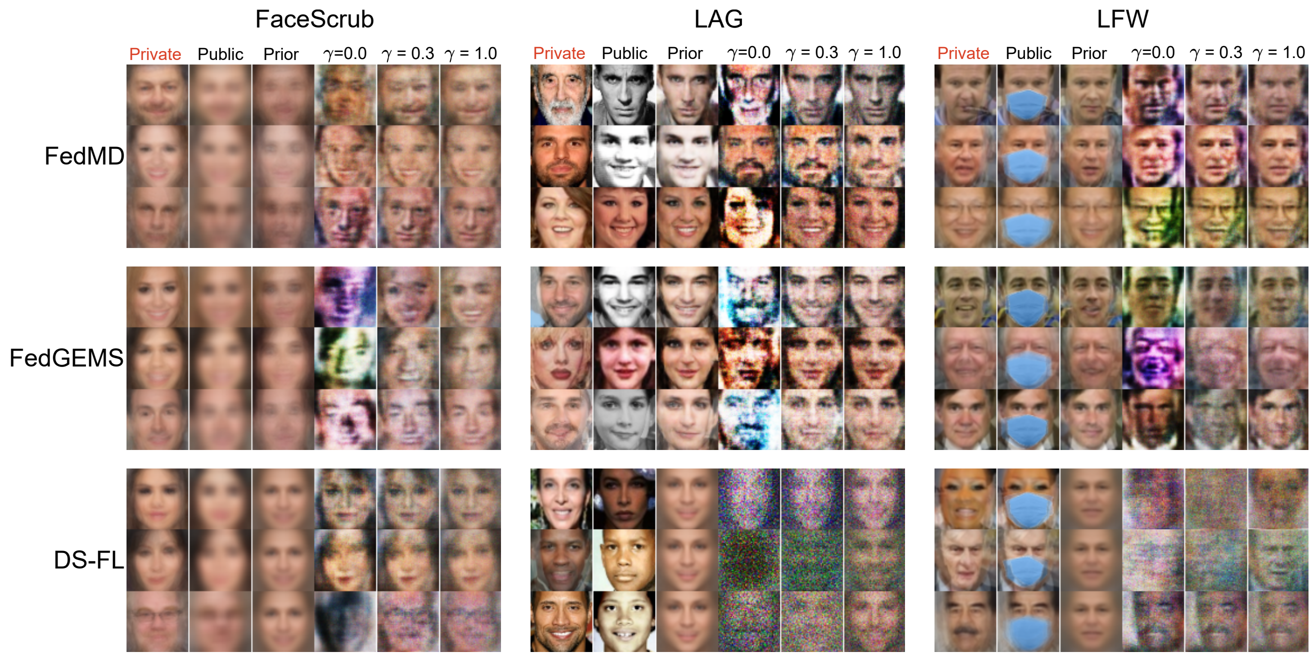

Figure S-1. Example of reconstructed images with various γ. Higher γ makes the recovered images closer to the prior images. Note that the attacker use the average of the public sensitive images as the prior for the unlabeled public dataset of DS-FL.

Dataset	FaceScrub			LAG			LFW		
Scheme	DS-FL	FedGEMS	FedMD	DS-FL	FedGEMS	FedMD	DS-FL	FedGEMS	FedMD
$\gamma = 0.00$	48.5	16.5	70.0	15.0	**39.0**	**65.5**	17.5	64.5	76.5
$\gamma = 0.03$	**62.5**	20.0	74.5	15.0	26.5	63.5	15.5	71.5	79.0
$\gamma = 0.10$	58.5	**27.0**	**75.0**	**17.5**	12.5	47.0	18.5	86.5	88.0
$\gamma = 0.30$	56.0	25.5	70.5	5.0	1.5	21.5	27.0	95.0	96.5
$\gamma = 1.00$	39.5	16.5	65.5	1.0	0.0	2.0	**45.5**	**100.0**	**100.0**

Table S-1. Attack accuracy with different γ. The magnitude of effective γ depends on the reliability of the quality of prior data (see also Tab. S-3).

Dataset	FaceScrub			LAG			LFW		
Scheme	DS-FL	FedGEMS	FedMD	DS-FL	FedGEMS	FedMD	DS-FL	FedGEMS	FedMD
$\gamma = 0.00$	0.457	0.359	0.532	0.163	0.286	0.354	0.298	0.457	0.445
$\gamma = 0.03$	0.468	0.408	0.560	0.163	0.313	0.357	0.313	0.522	0.476
$\gamma = 0.10$	0.487	0.467	0.590	0.170	0.338	0.376	0.320	0.554	0.525
$\gamma = 0.30$	0.528	0.570	0.648	0.184	0.362	0.397	0.340	0.617	0.623
$\gamma = 1.00$	**0.617**	**0.707**	**0.744**	**0.286**	**0.383**	**0.405**	**0.383**	**0.746**	**0.769**

Table S-2. SSIM between private and reconstructed images with different γ.

	FaceScrub	LAG	LFW
SSIM	0.860	0.483	0.923

Table S-3. SSIM between the private images and the prior images

recovered images with different τ against FedMD on LFW dataset, which indicates that larger τ gives better convergence. Fig. S-3 shows the reconstructed examples with different temperature τ. We can also observe the same trend in TBI (see Tab. S-6 and Tab. S-7).

Dataset	FaceScrub			LAG			LFW		
Scheme	DS-FL	FedGEMS	FedMD	DS-FL	FedGEMS	FedMD	DS-FL	FedGEMS	FedMD
$\tau = 0.3$	9.0	0.0	38.5	**32.5**	0.0	2.5	28.0	1.0	4.5
$\tau = 1.0$	16.5	0.0	64.0	31.5	1.0	14.5	**31.0**	3.5	24.5
$\tau = 3.0$	**62.5**	20.0	**74.5**	15.0	26.5	63.5	15.5	71.5	**79.0**
$\tau = 5.0$	52.0	**43.5**	**87.5**	3.0	**30.0**	**66.0**	14.5	**75.0**	71

Table S-4. Attack accuracy with different τ. The higher τ works when the labeled public dataset is available.

Dataset	FaceScrub			LAG			LFW		
Scheme	DS-FL	FedGEMS	FedMD	DS-FL	FedGEMS	FedMD	DS-FL	FedGEMS	FedMD
$\tau = 0.3$	**0.699**	**0.846**	**0.898**	**0.325**	0.533	0.543	**0.424**	0.767	**0.745**
$\tau = 1.0$	0.477	0.576	0.820	0.214	**0.554**	**0.572**	0.401	**0.771**	0.740
$\tau = 3.0$	0.468	0.408	0.560	0.163	0.313	0.357	0.313	0.522	0.476
$\tau = 5.0$	0.410	0.343	0.449	0.142	0.200	0.290	0.290	0.417	0.367

Table S-5. SSIM between private and reconstructed images with different τ.

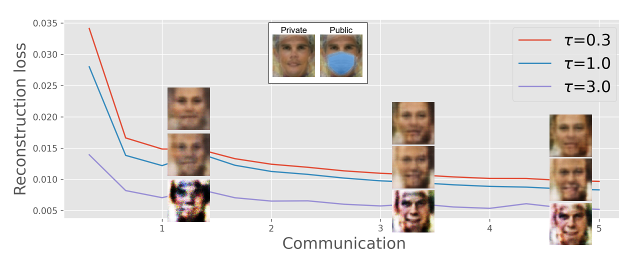

Figure S-2. Attack against FedMD in progress on LFW. Increasing τ makes convergence faster.

Impact of Public Dataset Size Since PLI relies on public knowledge, we also experiment with a smaller public

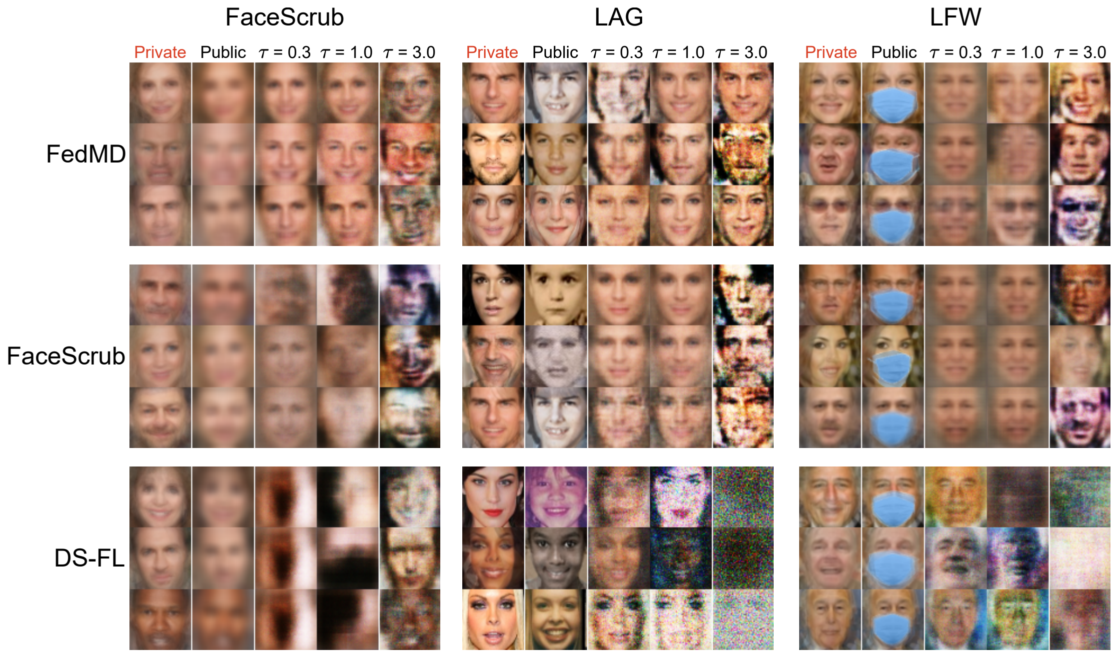

Figure S-3. Example of reconstructed images with various τ. Higher τ helps preserve the unique features of each individual but makes the reconstructed images noisier, especially for FedMD and FedGEMS. The effective τ is lower in some cases of DS-FL.

Dataset	FaceScrub			LAG			LFW		
Scheme	DS-FL	FedGEMS	FedMD	DS-FL	FedGEMS	FedMD	DS-FL	FedGEMS	FedMD
τ = 0.3	1.0	0.0	0.0	2.5	0.0	0.0	2.0	0.0	0.0
τ = 1.0	1.0	0.0	2.5	**7.5**	0.0	0.0	5.5	0.0	3.0
τ = 3.0	2.0	2.0	7.0	6.5	0.0	0.0	**17.5**	**9.5**	10.0
τ = 5.0	**3.5**	**3**	**8**	2.5	0	**1.5**	13	7.5	**16**

Table S-6. Attack accuracy of TBI with different τ. The trend is similar to the results of PIL.

Dataset	FaceScrub			LAG			LFW		
Scheme	DS-FL	FedGEMS	FedMD	DS-FL	FedGEMS	FedMD	DS-FL	FedGEMS	FedMD
τ = 0.3	**0.638**	**0.731**	**0.525**	**0.764**	**0.486**	**0.799**	**0.688**	**0.492**	**0.801**
τ = 1.0	0.536	0.489	0.436	0.539	0.480	0.792	0.495	0.464	0.780
τ = 3.0	0.371	0.367	0.421	0.360	0.205	0.593	0.346	0.213	0.492
τ = 5.0	0.375	0.358	0.343	0.312	0.144	0.444	0.405	0.151	0.384

Table S-7. SSIM of TBI between private and reconstructed images with different τ.

dataset to test its effect. While the default number of sensitive labels is 200, we set the number of clients to 10 and the number of sensitive labels to 500, which decreases the number of insensitive labels and the amount of public dataset. Tab. S-8 reports results on FedMD on this smaller public dataset, showing that both accuracy and quality decline. This indicates that a larger public dataset leads to more severe privacy violation.

	SSIM			Accuracy		
	FaceScrub	**LAG**	**LFW**	**FaceScrub**	**LAG**	**LFW**
Default size	0.560	0.357	0.476	74.5	63.5	79.0
Smaller size	0.251	0.295	0.370	72.6	47.4	73.4

Table S-8. Results with the smaller public dataset on FedMD with K=10. Smaller public dataset damages attack performance.

E Information Leakage

We compare logit-based attack with standard gradient-based attack via mutual information (MI). [3] finds that we can quantify information leakage between the input and output of a system by MI. We prove that the gradient w.r.t. the model's parameters has higher mutual information between input logits than output. Following Inequal. S-8 suggests that gradient can leak more information about the input than the output logit does.

Proposition S-1. *Let a neural network contain a biased fully-connected layer as the last layer with a differentiable activation function $y = h(Az+b)$, where h is the activation function; $y \in \mathbb{R}^{N_y}$, $z \in \mathbb{R}^{N_z}$, $A \in \mathbb{R}^{N_y \times N_z}$, $b \in \mathbb{R}^{N_y}$ are the output, input, weight and bias of the last layer, respectively. Then, if $\frac{\partial L}{\partial b}$ is not a zero vector, we have;*

$$I(x; \frac{\partial L}{\partial A}, \frac{\partial L}{\partial b}) \geq I(x; y), \tag{S-8}$$

where L is the differentiable loss function, x is the input data, and I denotes mutual information.

The bellow proof is based on Prop 3.1 in [2].

Proof. Since h is differentiable, we have the following equations;

$$\frac{\partial L}{\partial b_i} = \frac{\partial L}{\partial y_i}\frac{\partial y_i}{\partial b_i} = \frac{\partial L}{\partial y_i} h^{(1)}(Az+b)_i, \tag{S-9}$$

$$\frac{\partial y_i}{\partial A_i} = h^{(1)}(Az+b)_i z^T \tag{S-10}$$

, where i is the index of A's row. From the above equations, we can analytically determine z from $\frac{\partial L}{\partial b_i}$ and $\frac{\partial L}{\partial A_i}$ as follows;

$$z^T = \frac{1}{h^{(1)}(Az+b)}\frac{\partial y_i}{\partial A_i} \tag{S-11}$$

$$= \frac{1}{h^{(1)}(Az+b)}\frac{\partial y_i}{\partial L}\frac{\partial L}{\partial A_i} \tag{S-12}$$

$$= \frac{\partial L}{\partial A_i} / \frac{\partial L}{\partial b_i} \tag{S-13}$$

Then, if we think the neural network as a Markov chain $x \rightarrow z \rightarrow y$, the data processing inequality [1] leads to Inequal. S-8;

$$I(x; \frac{\partial L}{\partial A}, \frac{\partial L}{\partial b}) \geq I(x; z) \geq I(x; y) \tag{S-14}$$

□

F Gradient Inversion Attack

Algorithm S-4 Gradient inversion attack

Input: The number of communication T, the target model F, the number of clients K, the number of classes J, the number of classes of each private dataset $\{J_i\}_{i=1\dots C}$, the dimension of input d.

Output: Reconstructed data $\{X'_i \in \mathbb{R}^{d \times J_i}\}_{i=1\dots C}$

```
for t = 1 ← T do
    for i = 1 ← C do
        The server receives ∇W_i from client k.
        if t == 1 then
            Infer Y_i.
            X'_i ∈ R^{d×J_i} ← N(0,1)
        for m = 1 ← M do
            ∇W'_i ← ∂ℓ(f(X'_i, W_i), Y_i) / ∂W_i
            X'_i ← X'_i − η ∇_{X'_i} L_GB(X'_i)
return {X'_i}_{i=1...C}
```

Although the existing gradient inversion methods focus on reconstructing the exact batch data and labels, our interest is in recovering the class representation of the private training dataset. Then, we view that the received gradient ∇W_i is calculated with $X_i \in \mathbb{R}^{J_i \times d}$, where X_i represents the class representations of client k's private dataset, J_i is the number of unique classes of the dataset, and d is the dimension of the input data. The attacker can infer the labels used to train the local model from the received gradient with the batch label restoration method proposed in [4]. Then, we optimize dummy class representations $X'_i \in \mathbb{R}^{J_i \times d}$ with the following cost function;

$$L_{GB}(X'_i) = 1 - \frac{\langle \nabla W'_i, \nabla W_i \rangle}{||\nabla W'_i|| ||\nabla W_i||} + \gamma \mathrm{TV}(X'_i) \tag{S-15}$$

, where TV denotes the total variation and γ is its coefficient. This cost function is the same as the one used in [2]. Note that unlike our proposed attack against FedMD-like schemes, the attacker must know the number of unique labels in each local dataset in advance. In our experiments, we set γ to 0.01, and use Adam optimizer with a learning rate of 0.3.

Tab. S-9 reports the accuracy of gradient inversion attack against FedAVG for 10 clients. Note that this gradient inversion attack does not utilize the prior data based on the public dataset. Across all three datasets, the attack accuracy is higher than PLI without prior data (γ = 0.0), which indicates that gradients can potentially leak more private information than PLI without (see Prop. S-1 in Appendix E).

LFW	LAG	FaceScrub
85.5%	98.5%	95%

Table S-9. Gradient inversion attack against FedAVG ($K = 10$). Attack accuracy is higher than logit-based attacks on all datasets.

References

[1] Thomas M Cover. Elements of information theory. John Wiley & Sons, 1999. 5

[2] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? Advances in Neural Information Processing Systems, 33:16937–16947, 2020. 5

[3] Tianhao Wang, Yuheng Zhang, and Ruoxi Jia. Improving robustness to model inversion attacks via mutual information regularization. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 11666–11673, 2021. 4

[4] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16337–16346, 2021. 5