Breaching FedMD: Image Recovery via Paired-Logits Inversion Attack

Hideaki Takahashi\textsuperscript{*} \quad Jingjing Liu\textsuperscript{‡} \quad Yang Liu\textsuperscript{‡}

Abstract

Federated Learning with Model Distillation (FedMD) is a nascent collaborative learning paradigm, where only output logits of public datasets are transmitted as distilled knowledge, instead of passing on private model parameters that are susceptible to gradient inversion attacks, a known privacy risk in federated learning. In this paper, we found that even though sharing output logits of public datasets is safer than directly sharing gradients, there still exists a substantial risk of data exposure caused by carefully designed malicious attacks. Our study shows that a malicious server can inject a PLI (Paired-Logits Inversion) attack against FedMD and its variants by training an inversion neural network that exploits the confidence gap between the server and client models. Experiments on multiple facial recognition datasets validate that under FedMD-like schemes, by using paired server-client logits of public datasets only, the malicious server is able to reconstruct private images on all tested benchmarks with a high success rate.

1 Introduction

Federated Learning (FL) \cite{28} is a distributed learning paradigm, where each party sends the gradients or parameters of its locally trained model to a centralized server that learns a global model with the aggregated gradients/parameters. While this process allows clients to hide their private datasets, a malicious server can still manage to reconstruct private data (only visible to individual client) from the shared gradients/parameter, exposing serious privacy risks \cite{30,43,49}.

One effective solution against such attacks is Federated Learning with Model Distillation (FedMD) \cite{21}, where each party uses both its private data and a public dataset for local training and sends its predicted logits on the public dataset (termed public knowledge) to the server, who then performs aggregation and sends the aggregated logits back to each party for next iteration. Therefore, in FedMD, the server can only access the predicted logits on public dataset, thus preventing leakage from gradients or parameters and retaining data and model heterogeneity with low communication cost \cite{26}. Prior work \cite{4,14,18,26,29} suggests that FedMD is a viable defense to the original FL framework, and many studies \cite{4,6,14,21,24} consider FedMD as a suitable solution to solving real-life problems. This has catalyzed a broad application of FedMD-like schemes \cite{5,6} in industrial scenarios \cite{16,18,25,29,37,46,47}.

Despite its popularity, few studies have investigated the safety of sharing public knowledge in FedMD. Contrary to the common belief that FedMD is reliably safe, we found that it is actually vulnerable to serious privacy threats, where an adversary can reconstruct a party’s private data via shared public knowledge only. Such a reconstruction attack is nontrivial, since to guarantee that the adversary recovers its private data strictly from public knowledge only, the attack needs to follow two inherent principles: Knowledge-decoupling and Gradient-free. ‘Gradient-free’ means that the adversary can recover private data without access to gradients. ‘Knowledge-decoupling’ means that the attack ‘decouples’ private information from learned knowledge on public data only, and directly recovers private data. However, existing inversion attacks \cite{11,42,49} fail to meet the ‘knowledge-decoupling’ requirement as they do not consider the case where the target model is trained on both private and public datasets. Recent TBI \cite{42} proposed a gradient-free method with inversion network, but still relied on availability of private data (as shown in Tab. 1).

In this paper, we propose a novel Paired-Logits Inversion (PLI) attack that is both gradient-free and fully decouples private data from public knowledge. We observe that a local model trained on private data will produce more confident predictions than a server model that has not seen the private data, thus creating a "confidence gap" between a client’s logits and the server’s logits. Motivated by this, we design a logit-based inversion neural network that exploits this confidence gap between the predicted logits from an auxiliary server model and those from the client model (paired-logits). Specifically, we first train an inversion neural network model based on public dataset. The input of the
inversion network is the predicted logits of server-side and client-side models on the public data. The output is the original public data (e.g., raw image pixels in an image classification task). Then, through confidence gap optimization via paired-logits, we can learn an estimation of server and client logit for the target private data. To ensure the image quality of reconstructed data, we also propose a prior estimation algorithm to regularize the inversion network. Lastly, we feed those estimated logit to the trained inversion model to generate original private data. To the best of our knowledge, this is the first work to investigate the privacy weakness in FedMD-like schemes and to successfully recover private images from shared public knowledge with high accuracy.

We evaluate the proposed PLI attack on facial recognition task, where privacy risks are imperative. Extensive experiments show that despite the fact that logit inversion attacks are more difficult to accomplish than gradient inversion attacks (logits inherently contain less information than gradients), PLI attack successfully recovers original images in the private datasets across all tested benchmarks. Our contributions are three-fold: 1) we reveal a previously unknown privacy vulnerability of FedMD; 2) we design a novel paired-logits inversion attack that reconstructs private data using only the output logit of public dataset; 3) we provide a thorough evaluation with quantitative and qualitative analysis to demonstrate successful PLI attacks to FedMD and its variants.

Table 1. Comparison between PLI attack and existing inversion attacks. GF: Gradient-free. KD: Knowledge-decoupling.

<table>
<thead>
<tr>
<th>Method</th>
<th>Leak</th>
<th>GF</th>
<th>KD</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBI [42]</td>
<td>logit</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>DLG [49], GS [12]</td>
<td>gradient</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>dDLG [45], CPL [40]</td>
<td>parameter</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>mGAN-AI [39], Secret Revealer [44], GAN Attack [13]</td>
<td>logit</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

2 Problem Definition

In this study, we investigate the vulnerability of FedMD-like schemes and design successful attacks that can breach FedMD to recover private confidential data. As a case study, we choose image classification task, specifically face recognition, as our focus scenario. This is because in real applications (e.g., financial services, social networks), the leakage of personal images poses severe privacy risks. In this section, we first provide a brief overview of FedMD framework with notations. Then, we give a formal definition of image classification task under the federated learning setting.

FedMD [21] is a federated learning setting where each of K clients has a small private dataset \(D_k := \{(x_i^k, y_i^k)\}_{i=1}^{N_k}\), which is used to collaboratively train a classification model of \(J\) classes. There is also a public dataset \(D_p := \{(x_i^p, y_i^p)\}_{i=1}^{N_p}\), shared by all parties. Each client \(k\) trains a local model \(f_k\) with parameters \(W_k\) on either \(D_k\), \(D_p\) or both, and sends its predicted logits \(l^k_i := \{l_i^k\}_{i=1}^{N_p}\) on \(D_p\) to the server, where \(l_i^k = f_k(W_k; x_i^p)\). The server aggregates the logits received from all clients to form consensus logits \(l^p := \{l_i^p\}_{i=1}^{N_p}\). The consensus logits are then sent back to the clients for further local training (See the complete algorithm in Appendix B).

Several schemes follow FedMD with slight variations. For example, FedGEMS [6] proposes a larger server model \(f_0\) with parameters \(w_0\) to further train on the consensus logits and transfer knowledge back to clients. DS-FL [15] uses unlabeled public dataset and entropy reduction aggregation (see Appendix B for details). These frameworks only use public logits to communicate and transfer knowledge, which is the targeted setting of our attack.

Image Classification task can be defined as follows. Let \((x_t^a, y_t^a)\) denote image pixels and class label ID, respectively, and \(L\) denote the set of target labels \(\bigcup_{k=1}^{K} \{(y_i^k)\}_{i=1}^{N_k}\). We aim to use public logits \(l^k_i\) obtained for \(D_k\) to reconstruct private class representation \(x_j\) for any target label \(j \in L\). We further assume that \(D_a\) is made up of two disjoint subsets \(D_0 := \{(x_i^0, y_i^0)\}\) and \(D_a := \{(x_i^a, y_i^a)\}\), where \(D_0\) consists of public data of non-target labels, while \(D_a\) contains images from a different domain for all target and non-target labels. For example, in face recognition tasks, the feature space of \(D_a\) might include an individual’s insensitive images, such as masked or blurred faces (see Fig. 1). We also suppose the server is honest-but-curious, meaning it cannot observe or manipulate gradients, parameters, or architectures of local models.

![Figure 1. Examples from three datasets containing private and public data. Public dataset consists of two subsets: \(D_p = D_0 \cup D_a\). \(D_0\) and \(D_a\) consist of images of non-target and target labels (Unmasked (LFW), Adult (LAG) and Clean (FaceScrub)), while \(D_a\) contains images from a different domain (Masked (LFW), Young (LAG), Blurred (FaceScrub)).](image-url)
3 Paired Logits Inversion Attack

In this section, we first describe how to train the inversion neural network for private data recovery (Sec. 3.1), then explain how to estimate private logits by optimizing the confidence gap between the logits predictions of server and clients (Sec. 3.2). To prevent too much deviation from real data, we also introduce a prior estimation to regulate the inversion network training (Sec. 3.3). Lastly, we will explain how to recover private data using the trained inversion network, the estimated logits and the learned prior (Sec. 3.4).

3.1 Inversion Neural Network

Logit-based inversion is a model inversion attack that recovers the representations of original training data by maximizing the output logits w.r.t. the targeted label class of the trained model. This inversion typically requires access to model parameters, which is not accessible in FedMD. In order to perform a gradient-free inversion attack, [42] proposes a training-based inversion method (TBI) that learns a separate inversion model with an auxiliary dataset, by taking in output logits and returning the original training data. Inspired by this, we insert an inversion attack on the server side of FedMD. However, we show that logit-based inversion is more challenging than gradient-based inversion since logits inherently contain less information about the original data (See Appendix. E). To tackle the challenge, we train a server-side model \( f_0 \) on the public dataset with parameters \( W_0 \). The server-side logits are then updated as :

\[
\ell_i^0 = f_0(W_0; x_i^0)
\]

Next, we train an inversion model using client-server paired logits, \( \ell^k \) and \( \ell^0 \) on the public data subset \( D_0 \) only, denoted as \( G_0 \):

\[
\min_\theta \sum_i \left| \left| G_\theta(p_i^0, p_i^k) - x_i^0 \right|_2 \right|
\]

\[
p_i^0, \tau = \text{softmax}(\ell_i^0, \tau), \quad p_i^k, \tau = \text{softmax}(\ell_i^k, \tau)
\]

where \( \text{softmax}(\cdot, \tau) \) is the softmax function with temperature \( \tau \). Note the distribution gap between \( \ell^0 \) and \( \ell^k \) is the key driver for successfully designing such a paired-logits inversion attack, as will be demonstrated in Sec 3.2.

To enhance the quality of reconstruction, we leverage the auxiliary domain features \( D_a \) to obtain a prior estimation for each data sample \( \bar{x}_i \) via a translation algorithm (detailed in Sec. 3.3). The reconstruction objective is therefore summarized as:

\[
\min_\theta \sum_i \left| \left| G_\theta(p_i^0, p_i^k) - x_i^0 \right|_2 + \gamma \left| \left| G_\theta(p_i^0, p_i^k) - \bar{x}_i \right|_2 \right|
\]

The second term enforces the recovered data to be close to the prior estimation.

Since the server already has access to the public dataset and its corresponding output logits, it can train \( G_0^k \) to minimize Eq. 3 via backpropagation (line 6 in Algo 1). The architecture of the inversion model typically consists of transposed convolutional layers [8,42]. Notice that the inversion model is trained with public data only and did not observe any private data at training time. Once the inversion model is trained, it is able to reconstruct data in the public dataset.

However, the model cannot be used directly to reconstruct private data with sensitive labels yet, since it has never seen any true logits of the private data, the distribution of which is different from that of public data. Most existing inversion attacks [11–13, 39, 40, 44, 45, 49] require either gradients or parameters, thus not gradient-free. To reconstruct private data without gradients, we propose a new path for estimating the input logits of private data, by exploiting the confidence gap between server and clients, as below.

3.2 Confidence Gap Optimization

We observe that the server and client models exhibit a confidence gap on their predictions of the public data, mani-
fested by their predicted logits (Fig. 3). Specifically, the client model is more confident on the private dataset, resulting in a higher entropy of server logits. To exploit this, we propose a new metric for optimizing private logits, then anal-
alytically obtain the optimal solution, which is used as the input to inverted model for protection against FedMD.

Algorithm 1 PLI Attack

Input: The number of communication $T$, clients $K$, the set of target labels $L$, the inversion model $G^b_θ$, the translation model $A_φ$, the global model $f_0$, softmax temperature $τ$, and public dataset $D_p$.

Output: Class representations in the private dataset of $L$.

1: Generate prior $x_j$ for each $j \in L$ with $A_φ$. \(\triangleright\) Eq. 9
2: for $t = 1 \rightarrow T$ do
3: \hspace{1em} Receive $\{t_k^j\}$ from $k = 1..K$
4: \hspace{1em} for $k = 1 \rightarrow K$ do
5: \hspace{2em} Train $G^b_θ$ with Eq. 3 on $D_0$.
6: \hspace{2em} Train $G^θ_0$ with Eq. 10 for each $j \in L$.
7: \hspace{1em} Update global logits by aggregating $\{t_k^j\}$.
8: \hspace{1em} Train $f_0$ and send the global logits to each client.
9: for $j \in L$ do
10: \hspace{1em} Reconstruct $\{\hat{x}_j^k \leftarrow G^b_θ(p_0^j, p^k_j, τ)\}_{k=1}^K$
11: \hspace{1em} Pick the best from $\{\hat{x}_j^k\}$. \(\triangleright\) Eq. 11
12: return $\{\hat{x}_j\}_{j \in L}$

Recall that the private and public data domains for the target label $j$ are different. Since the server-side model is not directly trained on the private dataset, we can assume that $f_0$ returns less confident outputs on private data compared to $f_k$, in this sense $Q$ reinforces the knowledge-decoupling principle since $Q$ increases when the reconstructed data of $j$ is similar to the private data, and decreases when too close to the public data. For instance, $Q$ takes a lower value when the reconstructed data is masked, even if its label seems $j$.

**Analytical Solution** We can analytically solve Eq. 5 w.r.t. $p^k_j, τ$ and $p^0_j, τ$, and the optimal values for target label $j$ that maximizes $Q$, as follows:

$$\arg\max_{p^k_j, τ, p^0_j, τ} Q(x_j^k)$$

Detailed derivation can be found in Appendix A. Given an inversion model $G^b_θ$ that takes paired logits and returns the original input $x$, we have the following equation:

$$\arg\max_{x_j^k} Q(x_j^k) = G^b_θ(p^0_j, p^k_j, τ)$$

The empirical impact of feature space gap between public and private data is examined in Appendix E.

### 3.3 Prior Data Estimation

To prevent the reconstructed image from being unrealistic, we design a prior estimation component to regulate the inversion network. A naive approach is using the average of $D_0$ (public data with the same domain) as the prior data for any target label $i$:

$$\bar{x}_i = \frac{1}{|D_0|} \sum_{i \in D_0} x_i^0$$

Here the server uses the same prior for all the target labels.

When the public dataset is labeled, such as for FedMD and FedGEMS, the adversary can generate tuned prior for each label by converting $D_a$ with the state-of-the-art...
translation method such as autoencoder [7, 23, 34] and GAN [17, 19, 48]. Specifically, the server can estimate the prior data $\hat{x}_j$ for the target label $i$ with translation model $A_\phi$ as follows:

$$\hat{x}_j = \frac{1}{|D_a|} \sum_{x \in D_a} A_\phi(x)$$

(9)

### 3.4 Reconstruction of Private Data

Once the inversion model is trained with Eq. 3, the attacker uses the optimal logits (obtained from Sec. 3.2) to estimate the private data with $G_\theta$. To make the final prediction more realistic, the attacker further fine-tunes the inversion model by restricting the distance between the reconstructed image and their prior data (obtained from Sec. 3.3) in the same way as in Eq. 3:

$$\min_{\theta} \sum_{j \in L} \gamma || G_\theta^k(p_{j,\tau}, \hat{p}_{j,\tau}) - \hat{x}_j ||_2$$

(10)

Finally, given a reconstructed image from each client, the attacker needs to determine which of the $k$ images $\{\hat{x}_j\}_{k=1}^k$ from $k$ clients is the best estimation for target label $j$. We design the attacker to pick the most distinct and cleanest image as the best-reconstructed data based on: 1) similarity to groundtruth image via SSIM metric [38]; 2) data readability measured by Total Variation (TV) [35], which is commonly used as regularizer [12,43]. Specifically, the attacker chooses reconstructed private data $\hat{x}_j$ by:

$$\arg \min \sum_{k=1,k \neq k'} SSIM(\hat{x}_j^k, \hat{x}_j^{k'}) + \beta TV(\hat{x}_j^k)$$

(11)

where $\hat{x}_j^k = G_\theta^k(p_{j,\tau}, \hat{p}_{j,\tau})$. Lower SSIM indicates that the image is not similar to any other reconstructed privat pictures, and smaller TV means the image has better visual quality.

![SSIM distribution (left) and Reconstructed Images (right) for LAG.](image)

Figure 4. SSIM distribution (left) and Reconstructed Images (right) for LAG.

The reasoning for choosing Eq. 11 as the criterion is demonstrated in Fig. 4, where the left figure shows the distribution of the first term of Eq. 12 for clients who own the private data for the target label (blue) and clients who don’t (red). Fig. 4 also provides reconstructed examples from all clients, where $k = 3$ is the ground-truth client, whose recovered image has the (almost) lowest SSIM. This figure also demonstrates the importance of $\beta$ (the trade-off weight in Eq. 11, where a highly-noisy image ($k=8$) results in an even lower SSIM but is penalized by high TV score via a balanced weight ($\beta \geq 0.1$ in this case).

The complete algorithm is shown in Algo 1.

### 4 Experiments

We evaluate our proposed attack on face recognition task, which inherits high privacy risks in practice. We attack three schemes: FedMD [21], DS-FL [15], and FedGEMS [6] (see pseudo code in Appendix B). First, we describe detailed comparison between our approach and the baseline method TBI on various attacks. Then, we provide a thorough analysis on the impact of the learned prior, as well as the trade-off between image quality and attack accuracy in different attacks. Ablation studies further provide thorough analysis and insights on the effect of each PLI component and its comparison with gradient-based attack.

#### 4.1 Experimental Setup

**Datasets and Metrics** Three standard datasets are used in our experiments: Large-Age-Gap (LAG) [3], Masked LFW [33], and FaceScrub [31]. LAG [3] contains both youth and adult images of celebrities. Masked LFW [33] is created by automatically adding masks to the faces of images in the LFW dataset [20]. For LAG and LFW, we treat adult and non-masked images as sensitive features, and young and masked images as insensitive features. We use the images of 1000 subjects who have the highest numbers of photos, so the number of classes is 1000. Among these, we randomly pick 200 classes as private labels, and treat the remaining classes as public labels. For FaceScrub [31], we randomly select 330 persons and treat all their pictures as the public dataset. Then, we blur half of the images of the remaining 200 individuals with box kernel and merge them to the public dataset as auxiliary domain $D_a$. The private datasets consist of the remaining clean photos. All pictures of each dataset are resized to 64x64 and normalized to [-1, 1].

An attack is considered successful when:

$$SSIM(\hat{x}_j, x_j) > \max(SSIM(\hat{x}_j, x_{-j}), SSIM(\hat{x}_j, x_p))$$

where $\hat{x}_j$: averaged private image of label $j$, $x_{-j}$: averaged private image of each label except $j$, $x_p$: averaged public image of any label. Assume $M$ is the number of target labels with successful attack, and $N$ is the number of all target labels. **AttackAccuracy** is defined by $M/N$.

#### Attack Targets

We target three types of schemes: FedMD [21], DS-FL [15], and FedGEMS [6]. For sim-
plicity, the server and clients use the same neural network with a convolutional layer, a max-pooling layer, and a linear layer with ReLU (Code. 1 in Appendix B). DS-FL assumes that the public dataset is unlabeled, so the total number of classes in DS-FL equals the number of classes in the local dataset, 200. Classification loss is cross entropy for all schemes, and distillation loss is L1-loss for FedMD, KL divergence for FedGEMS, and cross-entropy for DS-FL (as in original papers). See Appendix C for the detailed hyperparameters.

For the translation model $A_g$ for learned prior, the attacker trains CycleGAN [48] for LAG and LFW, and DeblurGAN-v2 [19] for FaceScrub in advance on the public dataset when it is labeled (Eq. 9). CycleGan is used for facial aging and mask removing in some work [10, 32, 36], and DeblurGAN-v2 is one of the state-of-the-art methods to deblur images. For DS-FL, the server uses the average of public sensitive features as the prior (Eq. 8). For the inversion model, we adopt the same architecture used in TBI (see Code. 2 in Appendix B). Detailed hyper-parameters can be found in Appendix C.

### 4.2 Comparison with Baseline

Tab. 2 reports the attack accuracy of PLI in comparison with TBI. We use $\tau=3.0$ and $\gamma=0.03$ here as the empirically optimal setting across schemes and datasets. Results show that PLI outperforms TBI in all settings. Fig. 5 shows some reconstructed images from FedMD. The malicious party tries to recover private images. As the communication goes (higher $t$), the quality of reconstructed images improves (higher similarity to the groundtruth of private images). These images successfully capture the distinct sensitive features of the original private data, demonstrating that public logits can leak private information. On the other hand, the reconstructed images from TBI tend to be closer to the insensitive public features. More recovered examples from FedGEMS and DS-FL can be found in Appendix D.

In general, attack accuracy on FedGEMS is lower than that on the other two schemes. We hypothesize that this is because FedGEMS trains iteratively on both private and hard-labeled public data, compared to FedMD that trains each model on labeled public dataset only. DS-FL does not even use labels on public data. Thus, local models in FedGEMS are more affected by the public dataset compared to FedMD and DS-FL, and more difficult to accurately recover private data.

### 4.3 Analysis on Impact of Prior

We also discuss how the quality of prior data is related to optimal $\gamma$, the tunable weight for prior.

![Figure 6. Examples of Prior for the labeled public dataset. While we can obtain relatively good prior images for LFW, the prior pictures for LAG fail to estimate the change of the shape of faces. For FaceScrub, the prior images are still not sharp enough.](image)

Fig. 6 shows some qualitative examples of prior synthesized from the generator $A_g$ with the labeled public data. For LFW, CycleGAN successfully replaces the masked region with unmasked mouth. However, some distinct features under the mask, such as beard or lip color, are still not recovered. For LAG, faces in prior data look older than the public images, but are still more similar to private ones. Specifically, CycleGAN renders public images in adult appearances while keeping the original face outline. In addition, the recovered image is often too young or too old compared to the private image, since the attacker does not know the true age of the target person. For FaceScrub, DeblurGAN-v2 can deblur public images to some extent, but estimated prior data is still closer to public images.

![Figure 5. Examples of reconstructed images during training by our method where $t$ represents the number of communications (FS: FaceScrub), “TBI” represents the reconstructed results from TBI. Although TBI generates insensitive images, our PLI successfully reconstructs sensitive images as the learning progresses.](image)
Quality (SSIM)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>FaceScrub</th>
<th>LAG</th>
<th>LFW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheme</td>
<td>DS-FL</td>
<td>FedGEMS</td>
<td>FedMD</td>
</tr>
<tr>
<td>TBI ($K = 1$)</td>
<td>87.0</td>
<td>1.0</td>
<td>92.5</td>
</tr>
<tr>
<td>PLI ($K = 1$)</td>
<td>91.5</td>
<td>29.0</td>
<td>94.0</td>
</tr>
<tr>
<td>TBI ($K = 10$)</td>
<td>2.0</td>
<td>0.5</td>
<td>7.0</td>
</tr>
<tr>
<td>PLI ($K = 10$)</td>
<td>62.5</td>
<td>20.0</td>
<td>74.5</td>
</tr>
</tbody>
</table>

Table 2. Results on attack accuracy (%).

The influence of $\gamma$ is reasonable as increasing $\gamma$ makes the reconstructed images closer to prior images, which generally decreases the noise but also eliminates some distinctive features (such as beards) that the prior image does not have (see Fig. 8).

Figure 7. Trade-off between quality and accuracy with various $\gamma$. Dot size corresponds to the magnitude of $\gamma$. Larger $\gamma$ leads to high quality but ruins attack accuracy for LAG and FaceScrub, where the prior images are not close enough to the ground-truth private images. For LFW, the prior sufficiently resemble the private images, and increasing $\gamma$ improves both accuracy and quality.

Figure 8. Effect of $\gamma$ in FedMD (FS: FaceScrub) with fixed $\tau$. Higher $\gamma$ makes the reconstructed images noiseless, but sometimes drops distinctive elements (e.g., beard in the first row).

Figure 9. Trade-off between quality and accuracy with various $\tau$. Dot size is proportional to the magnitude of $\tau$. Larger $\tau$ leads to high accuracy but ruins quality for FedMD and FedGEMS, where the local models can use the labeled public dataset. The appropriate $\tau$ is lower for DSFL, where the public data is unlabeled and the predicted logits on the public data are more ambiguous.

Figure 10. Effect of $\tau$ in FedMD with fixed $\gamma$ (FS: FaceScrub). Higher $\tau$ preserves the characteristic features but damages readability (e.g., skin color in the second row).

4.4 Attack Accuracy vs. Image Quality

Experiments show that temperature $\tau$ controls the trade-off between attack accuracy and image quality. Fig. 9 shows the attack performance on various temperature $\tau$ with fixed $\gamma = 0.1$ (see Appendix D for the detailed numerical values). We observe that higher $\tau$ improves accuracy but damages quality. The influence of $\tau$ is consistent with previous work [15], suggesting that a higher $\tau$ amplifies non-highest scores of predicted logits while losing class information.

In summary, these experiments exhibit a trade-off between accuracy and quality, as optimizing attack accuracy...
does not guarantee improved quality. By finding a sweet spot (via $\gamma$ and $\tau$), we can empirically learn an optimal setting with high accuracy and acceptable quality.

### 4.5 Ablation Study

<table>
<thead>
<tr>
<th></th>
<th>FaceScrub</th>
<th>LAG</th>
<th>LFW</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_j^k$</td>
<td>55.5</td>
<td>57.0</td>
<td>71.5</td>
</tr>
<tr>
<td>$p_0^j + \alpha H(p^0)$</td>
<td>70.0</td>
<td>65.5</td>
<td>76.5</td>
</tr>
</tbody>
</table>

Table 3. Ablation studies on $Q$ (FedMD with $K=10$) in terms of attack accuracy. PLI (second row) outperforms using only client-side logits $p_j^k$ in terms of accuracy.

We further investigate the influence of each term of $Q$. Tab. 3 shows the results of ablation studies on FedMD of $K=10$, $\tau=3.0$ and $\gamma=0$, by gradually adding server and local logits of Eq. 4. Note that the second row is PLI, and the optimal logits for the first row is $(\hat{p}_j^k, \hat{p}_0^j) = (1, -)$. The results show that it is effective to add both $p_j^k$ and $\alpha H(p^0)$. This phenomenon validates the importance of taking into consideration the confidence gap between the server and client logits. The effects of public dataset size and magnitude of feature domain gap are reported in Appendix D. Appendix E and F also show that gradients lead to more severe privacy violations than public logits.

**Limitations** One limitation of our method is that it generates representative data of each class, not pixel-wise accurate image within the private dataset. Since the attacker cannot obtain anything directly calculated from the private dataset, reconstructing the exact image is still challenging.

### 5 Related Work

Federated Learning [28] is one of the representative algorithms for distributed learning. To overcome data and model heterogeneity, many studies focus on knowledge distillation [26]. In addition to the three schemes targeted in this work, there are several methods [22, 41] that communicate both distilled knowledge and gradients. While these schemes are the combination of gradient-based and knowledge-distillation-based federated learning, they are subject to existing gradient-based attacks, which are more severe attacks (as shown in Appendix E and F).

Model inversion attack aims to recover training data via machine learning models, which takes in information about the model (output logits, gradients, or parameters) and generates training data. Gradient inversion attack reconstructs the original training data by minimizing the distance between the received gradient that each client calculates on its local dataset and model and the gradient of synthesized data on the loss function, with model parameters used on the client side [12, 40, 45, 49]. Parameter-based inversion attacks utilize parameters of the target model. For example, [13, 39, 44] build GAN with built-in local models to recover the private data. Such attacks may not apply to FedMD scenarios when the server cannot access the parameters of client models. MI-FACE [11] reconstructs training data by directly optimizing synthesized data, so the learned model generates a high probability for the target class on that data. However, it requires the gradient of the model w.r.t synthesized data, thus not satisfying the gradient-free principle. In this sense, MI-FACE can be categorized as a combination of gradient-based and logit-based attacks.

### 6 Conclusion

This paper introduces an attack against FedMD and its variants that reconstructs private training data from the output logits of the model trained on both private and public datasets. Our proposed attack does not rely on gradients, parameters or model structures. Experiments on face recognition benchmarks demonstrate that confidential information can be recovered from output logits of public datasets. By demonstrating FedMD-like methods are not as safe as previously believed, we hope to inspire future design of more secure FL algorithms for diverse tasks. Potential future work includes adaptively choosing $\gamma$ based on the quality of the prior, and investigating protection mechanisms such as differential privacy [1,9] and homomorphic encryption [2,27].

### 7 Acknowledgement

This work was supported by the National Key R&D Program of China under Grant No.2022ZD0160504, Tsinghua-Toyota Joint Research Institute inter-disciplinary Program, and Xiaomi AI Innovation Research under grant No.202-422-002.
References


