A PROOF OF THEOREM 1

Theorem 1. *Let f_1 and f_2 are two models such that for any fixed label $y \in \mathcal{Y}$, $U_{f_1}(x_{ns}, y) \geq U_{f_2}(x_{ns}, y)$. Then, $S_{KL}(p(X_s|y, x_{ns})||p_{f_1}(X_s|y, x_{ns})) \geq S_{KL}(p(X_s|y, x_{ns})||p_{f_2}(X_s|y, x_{ns}))$.*

Proof. We can expand the KL divergence $D_{\text{KL}}(p(X_s|y, x_{ns})||p_{f_1}(X_s|y, x_{ns})$ as follows.

$$D_{\text{KL}}(p(X_s|y, x_{ns})||p_{f_1}(X_s|y, x_{ns})) \tag{1}$$
$$= \mathbb{E}_{X\sim p(X_s|y,x_{ns})}[\log p(X_s|y, x_{ns})] - \mathbb{E}_{X\sim p(X_s|y,x_{ns})}[\log p_{f_1}(X_s|y, x_{ns})] \tag{2}$$

Thus,

$$D_{\text{KL}}(p(X_s|y, x_{ns})||p_{f_1}(X_s|y, x_{ns})) - D_{\text{KL}}(p(X_s|y, x_{ns})||p_{f_2}(X_s|y, x_{ns})) \tag{3}$$
$$= \mathbb{E}_{X\sim p(X_s|y,x_{ns})}[\log p_{f_2}(X_s|y, x_{ns}) - \log p_{f_1}(X_s|y, x_{ns})] \tag{4}$$
$$= \sum_x p(X_s|y, x_{ns})\big(\log \frac{p_{f_2}(y|X_s, x_{ns})p(X_s|x_{ns})}{p_{f_2}(y|x_{ns})} - \log \frac{p_{f_1}(y|X_s, x_{ns})p(X_s|x_{ns})}{p_{f_1}(y|x_{ns})}\big) \tag{5}$$
$$= \sum_x p(X_s|y, x_{ns})\Big(\big(\log p_{f_2}(y|X_s, x_{ns}) - \log p_{f_2}(y|x_{ns})\big) - \big(\log p_{f_1}(y|X_s, x_{ns}) - \log p_{f_1}(y|x_{ns})\big)\Big) \tag{6}$$
$$= U_{f_2}(x_{ns}, y) - U_{f_1}(x_{ns}, y) \leq 0 \tag{7}$$

□

B EXPERIMENTAL DETAILS

B.1 NETWORK ARCHITECTURE

The GAN architectures for the GMI attacks without auxiliary knowledge, with corrupted private image, and with blurred private image, are shown in Figure 1, 2, and 3, respectively. Moreover, in the experiments, we use the same GAN architectures for the PII baseline and the GMI attacks.

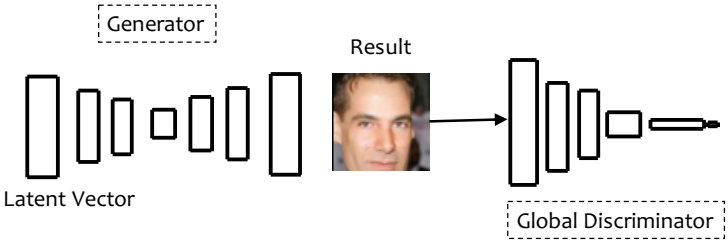

Figure 1: The GAN architecture for the attack without auxiliary knowledge.

The detailed architecture designs of the two encoders, the decoder of the generator, the local discriminator, and the global discriminator are presented in Table 1, Table 2, Table 3, Table 4, and Table 5, respectively.

The information of some network architectures used in the experiment section but not covered in the main text is elaborated as follows: (1) `LeNet` adapted from (Lecun et al., 1998), which has three convolutional layers, two max pooling layers and one FC layer; (2) `SimpleCNN`, which has five convolutional layers, each followed by a batch normalization layer and a leaky ReLU layer; (3) `SoftmaxNet`, which has only one FC layer.

B.2 THE DETAILED SETTING OF THE EXPERIMENTS ON "ATTACKING DIFFERENTIALLY PRIVATE MODELS"

We split the `MNIST` dataset into the private set used for training target networks with digits $0 \sim 4$ and the public set used for distilling prior knowledge with digits $5 \sim 9$. The target network is

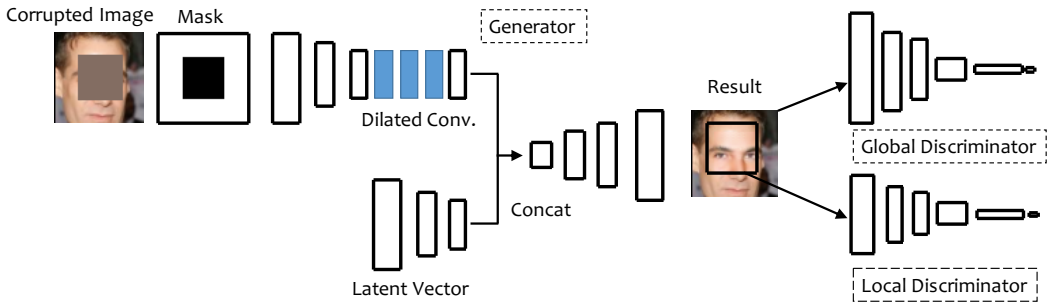

Figure 2: The GAN architecture for the attack with the auxiliary knowledge of a corrupted private image.

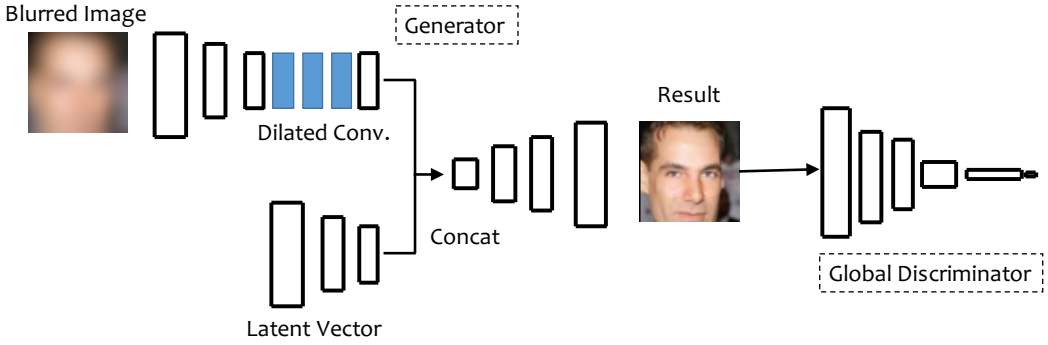

Figure 3: The GAN architecture for the attack with the auxiliary knowledge of a blurred private image.

Table 1: When the auxiliary knowledge is a corrupted private image, the upper encoder of the generator takes as input the corrputed RGB image and the binary mask. When the auxiliary knowledge is a blurred private image, the upper encoder only takes an image as input.

Type	Kernel	Dilation	Stride	Outputs
conv.	5x5	1	1x1	32
conv.	3x3	1	2x2	64
conv.	3x3	1	1x1	128
conv.	3x3	1	2x2	128
conv.	3x3	1	1x1	128
conv.	3x3	1	1x1	128
conv.	3x3	2	1x1	128
conv.	3x3	4	1x1	128
conv.	3x3	8	1x1	128
conv.	3x3	16	1x1	128

Table 2: The lower encoder of the generator that takes as input the latent vector.

Type	Kernel	Stride	Outputs
linear			8192
deconv.	5x5	1/2 x 1/2	256
deconv.	5x5	1/2 x 1/2	128

implemented as a Multilayer Perceptron with 2 hidden layers, which have 512 and 256 neurons, respectively. The evaluation classifier is a convulutional neural network with three convolution layers,

Table 3: The decoder of the generator.

Type	Kernel	Stride	Outputs
deconv.	5x5	1/2 x 1/2	128
deconv.	5x5	1/2 x 1/2	64
conv.	3x3	1x1	32
conv.	3x3	1x1	3

Table 4: The global discriminator.

Type	Kernel	Stride	Outputs
conv.	5x5	2x2	64
conv.	5x5	2x2	128
conv.	5x5	2x2	256
conv.	5x5	2x2	512
conv.	1x1	4x4	1

Table 5: The local discriminator. This discriminator only appears in the attack with the knowledge of a corrupted image.

Type	Kernel	Stride	Outputs
conv.	5x5	2x2	64
conv.	5x5	2x2	128
conv.	5x5	2x2	256
conv.	1x1	4x4	1

followed by two fully-connected layers. It is trained on the entire MNIST training set and can achieve 99.2% accuracy on the `MNIST` test set.

Differential privacy of target networks is guaranteed by adding Gaussian noise to each stochastic gradient descent step. We use the moment accounting technique to keep track of the privacy budget spent during training (Abadi et al., 2016). During the training of the target networks, we set the batch size to be 256. We fix the number of epochs to be 40 and clip the L2 norm of per-sample gradient to be bounded by 1.5. We set the ratio between the noise scale and the gradient clipping threshold to be $0, 0.694, 0.92, 3, 28$, respectively, to obtain the target networks with $\varepsilon = \infty, 9.89, 4.94, 0.98, 0.10$ when $\delta = 10^{-5}$. For model with $\varepsilon = 0.1$, we use the SGD with a small learning rate 0.01 to ensure stable convergence; otherwise, we set the learning rate to be 0.1.

The architecture of the generator in Section B.1 is tailored to the `MNIST` dataset. We reduce the number of input channels, change the size of kernels, and modify the layers of discriminators to be compatible with the shape of the `MNIST` data. To train the GAN in the first stage of our GMI attack, we set the batch size to be 64 and use the Adam optimizer with the learning rate 0.004, $\beta_1 = 0.5$, and $\beta_2 = 0.999$ (Kingma and Ba, 2014). For the second stage, we set the batch size to be 64 and use the SGD with the Nesterov momentum that has the learning rate 0.01 and momentum 0.9. The optimization is performed for 3000 iterations.

REFERENCES

M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278 – 2324, 12 1998. doi: 10.1109/5.726791.