

A Details of Experimental Settings and Additional Results

A.1 Detailed Extraction Attack Setting.

For the surrogate model training, we use SGD optimizer with a learning rate of 0.02 for 200 epochs. The learning rate is multiplied by a factor of 0.2 at epochs 60, 120 and 160. For Craft-ME, we craft an equal number of small-loss instances for each class. We use the Adam optimizer with a learning rate of 0.1 and set the total number of steps (iterations) to 20 or 50 to craft each image. For GAN-ME, we use a conditional-GAN model as the generator, its detailed architecture is given in Appendix A.11. For the generator training, we use Adam optimizer with a learning rate of $1e-4$ and apply the divergence-aware regularization [Yang et al., 2019] with a factor of 50 to mitigate the mode collapse problem. For GM-ME, we query with the entire training dataset (i.e. 50K) of each auxiliary dataset (CIFAR-10, SVHN and MNIST). For Train-ME and SoftTrain-ME, we apply standard data augmentation techniques including random rotating and horizontal flipping, during the surrogate model training. For SoftTrain-ME, we train the surrogate model with both the hard labels and gradient-based soft labels with α parameter of 0.9. Data augmentation is disabled if training uses soft labels.

A.2 Illustration of the attack workflow

The workflow of five proposed attacks are illustrated in Fig. 5. During SFL training, a malicious client (the attacker) can deploy either of five ME attacks depending on its data assumption.

For the no-data case, the attacker can use either Craft-ME or GAN-ME. Here the gradients are used to generate crafted data or update the conditional GAN as data generator. These two attacks require a longer preparation step, where the crafted data and data generator are derived, prior to the surrogate model training.

For all five attacks, the surrogate model is trained using the known client-side model as the initial model. For Train-ME, Craft-ME and GAN-ME, the standard cross-entropy loss is used. For GM-ME, the gradient matching loss is used, and for SoftTrain-ME, the gradient-based soft labels are used.

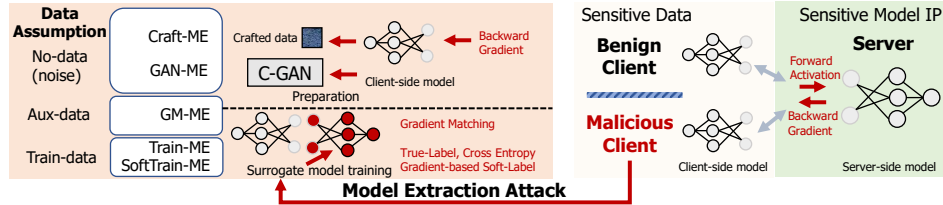


Figure 5: Detailed illustration of five proposed Model Extraction (ME) attacks in SFL. After preparation phase (if necessary), the attack completes by training the surrogate model till its convergence.

A.3 Gradient-based Attack Performance with Consistent Gradient Access

We did extensive experiments for ME attacks in different settings with consistent gradient query access and present the results here. These are in addition to what was presented in Section 5.1. The query budget is set at 1K, 10K and 100K. Results for all five ME attacks with different settings are shown separately in Fig. 6 (a), (b), (c), (d) and (e). The victim VGG-11 model has 91.89% validation accuracy on CIFAR-10 dataset. For GM-ME, we use CIFAR-100, SVHN, and MNIST as the auxiliary dataset.

Conclusion. We observe all ME attacks are equally successful for small N . Among different settings, Craft-ME performs better with 20 steps compared to 50 steps. This is possibly because for the same query budget, fewer steps results in more images being crafted. GAN-ME performance is much better with a larger query budget since a generator model needs more iterations of training to converge. GM-ME's performance heavily depends on the similarity of the auxiliary dataset. Because the victim model is on CIFAR-10, it performs well when CIFAR-100 is set as the auxiliary dataset while performing badly when MNIST is used. Moreover, attacks with training data perform much better than ME attacks without training data. Compared to Train-ME, SoftTrain-ME achieves better accuracy and fidelity when $N \geq 6$.

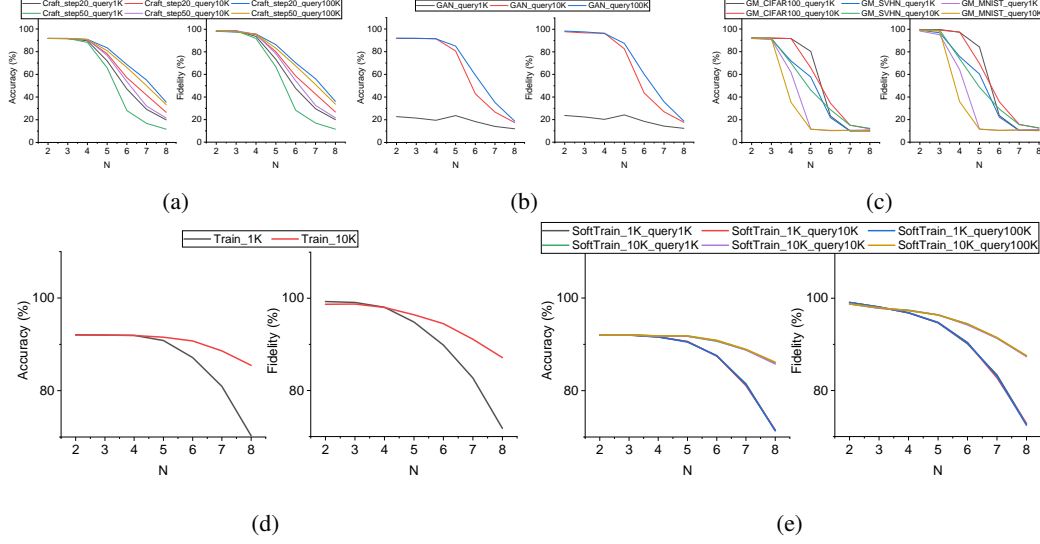


Figure 6: Additional results for **consistent** gradient query case. **Top row:** ME attacks without training data with different settings. (a) Craft-ME with different number of crafting steps and query budgets. (b) GAN-ME with different query budgets. (c) GM-ME with different auxiliary datasets and query budgets. **Bottom row:** ME attacks with training data with different settings. (d) Train-ME with 1K/10K training data. (e) SoftTrain-ME with 1K/10K training data with different query budgets.

522 A.4 Gradient-based Attack Performance with Inconsistent Gradient Access.

523 Here, we provide results for ME attacks for training-from-scratch settings with inconsistent gradient
524 query access; a subset of these results was presented in Section 5.2. We launch the attack by feeding
525 malicious inputs at late epochs, specifically, epochs 120 or 160, for the case when the number of
526 training epochs is 200. The attacker starts to collect gradients after the attack is launched till the
527 the end of training (epoch 200). We start the gradient collection from later epochs since by then
528 the model has achieved near-optimal accuracy and hence is valuable as an attack target. Also the
529 model updating is slower because of application of learning rate decay to make the gradients more
530 consistent.

531 In multi-client SFL, the original 50K training data is divided equally to 5 or 10 benign clients, denoted
532 as “5-client” and “10-client” case, respectively. The attacker is an additional client without training
533 data so a 5-client SFL really has 6 clients (5 benign clients and 1 malicious client). All clients,
534 including the attacker, perform an equal number of queries in each epoch. The performance of five
535 ME attacks with inconsistent gradient queries are shown separately in Fig. 7 (a), (b), (c), (d) and
536 (e). Because of the poisoning effect, the final model accuracy of the victim model is reduced by 2 ~
537 3%. For GM-ME, we use CIFAR-100 as the auxiliary dataset, and we only use the latest gradients to
538 perform gradient matching instead of using all collected gradients. We use “late50” to denote only
539 gradients collected in 50 latest training steps are used. This restriction greatly reduces the number of
540 gradients being available but makes them much more consistent.

541 **Conclusion.** Attacks without training data (Craft, GAN, GM MEs) work poorly with inconsistent
542 gradient queries. For Craft-ME, taking 20 steps also seems to work better in both 5-client and 10-
543 client cases. Collecting gradients starting later at epoch 160 gets better performance than starting early
544 at epoch 120 because of more consistent gradients. The starting-later rule also holds for GAN-ME
545 and GM-ME, where we can see starting later achieves consistently better ME attack performance.
546 For GM-ME, it only gets meaningful accuracy if only the latest gradients (within 10 training steps
547 to the end of training) are used, showing that it is extremely sensitive to gradient consistency. For
548 attacks with training data, we notice Train-ME attack performance is not affected because it does not
549 rely on gradients. However, SoftTrain-ME performs much worse because of the poisoning effect and
550 inconsistent gradients.

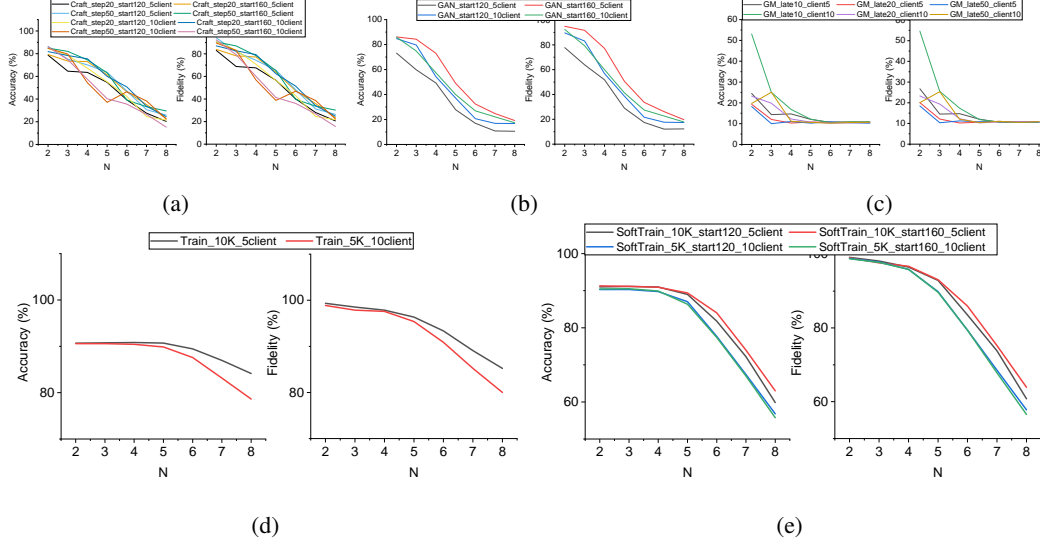


Figure 7: Additional results for **inconsistent** gradient query case. **Top row:** ME attacks without training data with different settings. (a) Craft-ME with different steps and starting epochs. (b) GAN-ME with different starting epochs. (c) GM-ME uses the latest gradients with different restrictions. **Bottom row:** ME attacks with training data with different settings. (d) Train-ME with 10K/5K training data. (e) SoftTrain-ME with 10K/5K training data with different starting epochs.

551 A.5 Accuracy Impact of Defensive Methods

552 We provide original accuracy, ME attack performance, as well as model inversion attack performance
 553 in addition to what was presented in the main paper in section 6. As shown in Table 3, L1 regular-
 554 ization works well for $N = 5$ where it reduces extraction performance a lot while slightly affecting
 555 original accuracy, and at the same time also improves the resistance to model inversion attack (better
 556 data privacy).

Table 3: Detailed defensive performance of L1 regularization (L1Reg) of VGG-11 model on CIFAR-10. Extraction performance of Train-ME with 1K training data is shown. Resistance to model inversion attack is shown by MSE.

| Regularization | Strength | N=4 | | | | N=5 | | | |
|----------------|----------|-------------|----------|----------|--------|-------------|----------|----------|--------|
| | | Orig. Accu. | Accuracy | Fidelity | MSE | Orig. Accu. | Accuracy | Fidelity | MSE |
| None | 0.0 | 91.45 | 91.02 | 96.94 | 0.0217 | 91.71 | 90.23 | 94.73 | 0.0114 |
| L1Reg | 5e-5 | 90.66 | 89.44 | 95.03 | 0.0274 | 90.43 | 87.45 | 91.10 | 0.0270 |
| L1Reg | 1e-4 | 87.90 | 86.24 | 93.68 | 0.0280 | 88.37 | 82.18 | 86.01 | 0.0239 |
| L1Reg | 2e-4 | 82.96 | 80.56 | 89.98 | 0.0262 | 85.00 | 76.45 | 80.78 | 0.0145 |

557 A.6 Non-IID Performance.

558 We demonstrate the ME attack performance in a non-IID setting, where the attacker only has access
 559 to training data from a subset of classes (C). The new set of results corresponding to Train-ME attack
 560 are shown in Table 4. We observe for C smaller than 5, attacker performance degrades badly for both
 561 CIFAR-10 and CIFAR-100 datasets.

562 A.7 Adversarial Attack Performance.

563 We demonstrate that with proper model IP protection, adversarial attacks can be mitigated. We assume
 564 the attacker uses the strongest Train-ME attack (with 1K data) to obtain a high-fidelity surrogate model
 565 to perform transfer adversarial attacks on the victim model with different IP protection strengths
 566 (SFL with different N). We use FGSM [Goodfellow et al., 2014], and targeted-PGD attack [Madry
 567 et al., 2017] to perform the transfer adversarial attack. We set the e for FGSM at 0.1, and PGD-target

Table 4: Model extraction performance of Train-ME attack on VGG-11 model on CIFAR-10 and CIFAR-100 dataset with Original Accuracy of 91.89% and 68.64%, respectively.

| Method | CIFAR-10 Accuracy | | | CIFAR-100 Accuracy | | |
|--------|-------------------|-------|-------|--------------------|-------|-------|
| | N=2 | N=5 | N=8 | N=2 | N=5 | N=8 |
| C = 1 | 47.58 | 46.75 | 38.42 | 6.79 | 6.79 | 6.13 |
| C = 2 | 82.45 | 79.30 | 58.69 | 13.18 | 13.36 | 11.25 |
| C = 5 | 91.70 | 88.90 | 65.70 | 32.77 | 29.65 | 17.90 |

at 0.002 for 50 iterations (the attacker randomly chooses the original and target label). We report the average Attack Success Rate (ASR) - the percentage of samples that are transferred successfully - to show the attacking performance. The new set of results is shown in Table 5. We see that both adversarial attacks achieve very high ASR for small N , where model IP protection is weak. On a SFL scheme with large N , adversarial attack performance degrades significantly using the surrogate model with less fidelity.

Table 5: Adversarial Attack ASR performance based on the surrogate model obtained using Train-ME attack, on VGG-11 model on CIFAR-10 with different N setting.

| Attack | Number of Server-side Layer (N) | | | | | | |
|------------|-------------------------------------|------|------|------|------|------|------|
| | N=2 | N=3 | N=4 | N=5 | N=6 | N=7 | N=8 |
| FGSM | 82.7 | 82.3 | 77.9 | 77.3 | 63.1 | 56.9 | 37.7 |
| PGD-target | 100 | 100 | 99.8 | 100 | 99.5 | 73.4 | 34.2 |

A.8 Surrogate Architecture Performance.

To investigate the impact on model extraction attacks caused by the surrogate model’s architecture difference, we designed four variants of the true server-side model, and used them as surrogate model architecture to perform model extraction attacks. We fixed the settings to $N = 5$ SFL and consistent gradient query budget to 10K. The new set of results are shown in Table 6 for VGG-11 model on CIFAR-10 dataset. For most attacks, architecture does not make a huge difference, and longer or wider surrogate architecture can achieve even better accuracy and fidelity. The exception is GM-ME, which achieves much higher extraction performance with the surrogate model having the same architecture.

Longer Architecture. Surrogate model has one extra fully connected layer compared to the original true server-side model.

Shorter Architecture. Surrogate model has one less fully connected layer compared to the original true server-side model.

Wider Architecture. Surrogate model has channel size that is 2 times of the original true server-side model

Thinner Architecture. Surrogate model has channel size half of the original channel size of the true server-side model.

A.9 Other Empirical Results.

In this section, we present more empirical results for ME attacks without training data to show that our claims can generalize to other architecture and datasets. The list of experiments are:

- 1. ME attack performance (without training data only) of VGG-11 on CIFAR-100 (Table 7). An interesting observation is GAN-ME performs worse than Craft-ME for consistent gradient cases for the increasing number of classes (100) makes the generator even harder to converge. While for the inconsistent gradient case, GAN-ME performs much better than Craft-ME because its generator can adapt to the inconsistent gradients and Craft-ME cannot.

Table 6: Extraction attack performance on surrogate models having slightly different architectures from the true architecture of the server-side model. N is fixed at 5, gradients are consistent and the query budget is 10K.

| Attacks | Accuracy (%) | | | | | Fidelity (%) | | | | |
|--------------|--------------|--------|---------|-------|---------|--------------|--------|---------|-------|---------|
| | same | longer | shorter | wider | thinner | same | longer | shorter | wider | thinner |
| Craft-ME | 76.67 | 75.05 | 77.90 | 79.00 | 74.86 | 78.38 | 76.70 | 79.72 | 81.04 | 74.74 |
| GAN-ME | 80.57 | 75.95 | 76.66 | 74.27 | 65.69 | 82.66 | 78.13 | 78.54 | 76.11 | 67.58 |
| GM-ME | 65.77 | 11.41 | 18.04 | 14.77 | 14.42 | 69.60 | 11.22 | 18.61 | 14.90 | 14.35 |
| Train-ME | 90.82 | 90.33 | 90.76 | 90.72 | 90.10 | 94.84 | 94.47 | 94.84 | 94.79 | 93.94 |
| SoftTrain-ME | 90.57 | 90.43 | 90.66 | 90.62 | 90.10 | 94.76 | 94.62 | 94.84 | 94.59 | 94.13 |

- 2. ME attack Performance (without training data only) of Vgg11 on 5% subset of FEMNIST dataset (62-class), following the same setting as leaf benchmark Caldas et al. [2018]’ online document (Table 8). We observe a similar trend as in VGG-11 on CIFAR-10 experiments.
- 3. ME attack Performance (without training data only) of MobileNetV2 on CIFAR-10 (Table 9). We observe a similar trend as in VGG-11 on CIFAR-10 experiments.

Table 7: Model extraction performance of gradient-based ME attacks with consistent gradient query (100K query budget) and inconsistent gradient query for 10-client SFL on **VGG-11 model CIFAR-100 dataset**. Original Accuracy is 68.64%. We use 20 crafting steps for the Craft-ME for both cases. For the inconsistent case, we launch ME attack at epoch 160, and use the “late10” setting for GM-ME.

| Case | Method | Accuracy (%) | | | | | Fidelity (%) | | | | |
|--------------------|----------|--------------|-------|-------|-------|--|--------------|-------|-------|-------|--|
| | | N=2 | N=3 | N=4 | N=5 | | N=2 | N=3 | N=4 | N=5 | |
| Fine-tuning | Craft-ME | 66.44 | 64.68 | 35.37 | 15.4 | | 86.97 | 81.37 | 40.35 | 16.7 | |
| | GAN-ME | 56.54 | 46.53 | 13.11 | 6.69 | | 69.91 | 55.56 | 14.86 | 7.11 | |
| | GM-ME | 68.76 | 68.4 | 57.87 | 1.28 | | 99.11 | 94.46 | 71.5 | 1.26 | |
| Train-from-scratch | Craft-ME | 11.53 | 8.49 | 2.61 | 2.41 | | 13.67 | 10.15 | 2.71 | 2.45 | |
| | GAN-ME | 49.4 | 41.9 | 22.1 | 10.75 | | 60.04 | 49.29 | 25.46 | 12.55 | |
| | GM-ME | 4.05 | 1.47 | 1.37 | 1.23 | | 4.92 | 1.79 | 1.54 | 1.19 | |

Table 8: Model extraction performance of gradient-based ME attacks with consistent gradient query (100K query budget) on **VGG-11 model FEMNIST dataset**. Original Accuracy is 74.62%. We use 50 crafting steps for the Craft-ME for both cases.

| Case | Method | Accuracy (%) | | | | | Fidelity (%) | | | | |
|-------------|--------------|--------------|-------|-------|-------|--|--------------|-------|-------|-------|--|
| | | N=2 | N=3 | N=4 | N=5 | | N=2 | N=3 | N=4 | N=5 | |
| Fine-tuning | Craft-ME | 53.20 | 43.57 | 43.04 | 40.50 | | 59.28 | 48.10 | 46.58 | 42.11 | |
| | GAN-ME | 10.59 | 7.19 | 5.27 | 4.02 | | 11.39 | 7.35 | 5.20 | 3.80 | |
| | GM-ME | 56.67 | 22.53 | 9.87 | 3.78 | | 69.70 | 25.14 | 10.10 | 3.68 | |
| | Train-ME | 70.32 | 68.47 | 68.80 | 67.70 | | 82.56 | 77.52 | 75.61 | 71.97 | |
| | SoftTrain-ME | 75.70 | 74.93 | 74.42 | 74.46 | | 83.87 | 81.24 | 77.39 | 76.30 | |

604 A.10 Time Cost Evaluation.

605 We evaluate time cost of five attacks on VGG-11 CIFAR-10 model (fine-tuning case). The time cost
606 measurement is done on a PC with a R7-5800X CPU and a single RTX-3090 GPU.

607 Table 10 provides time cost breakdown for two phases, namely, preparation phase and training the
608 surrogate model phase. The preparation phase includes crafting inputs in Craft-ME, fitting conditional
609 GAN in GAN-ME, and crafting soft labels in SoftTrain-ME. From the results, we can see the Craft-
610 ME needs the most preparation time and GAN-ME ranks the second. Both require generating crafted
611 data and training the generator using collected gradients. For training the surrogate model, GM-ME

Table 9: Model extraction performance of gradient-based ME attacks with consistent gradient query (100K query budget) and inconsistent gradient query for 10-client SFL on **MobilenetV2 model CIFAR-10 dataset**. Original Accuracy is 93.82%. We use 20 crafting steps for the Craft-ME for both cases. For the inconsistent case, we launch ME attack at epoch 160, and use the “late10” setting for GM-ME.

| Case | Method | Accuracy (%) | | | | Fidelity (%) | | | |
|--------------------|----------|--------------|-------|-------|-------|--------------|-------|-------|-------|
| | | N=2 | N=3 | N=4 | N=5 | N=2 | N=3 | N=4 | N=5 |
| Fine-tuning | Craft-ME | 92.29 | 76.74 | 72.74 | 61.08 | 96.04 | 77.86 | 73.24 | 61.46 |
| | GAN-ME | 92.67 | 79.17 | 68.92 | 57.61 | 96.46 | 80.35 | 69.7 | 58.25 |
| | GM-ME | 93.2 | 92.82 | 92.39 | 91.86 | 97.83 | 96.87 | 95.74 | 94.74 |
| Train-from-scratch | Craft-ME | 78.55 | 63.04 | 61.77 | 58.76 | 80.8 | 64.87 | 63.35 | 60.07 |
| | GAN-ME | 77.08 | 38.2 | 35.08 | 32.49 | 79.43 | 38.87 | 35.6 | 33.11 |
| | GM-ME | 31.23 | 11.25 | 15.25 | 17.81 | 32.73 | 11.46 | 15.14 | 17.9 |

Table 10: Time costs of proposed five attacks of attacking VGG-11 on CIFAR-10 (N=8) in fine-tuning case.

| Time Cost (s) | Craft | GAN | GM | Train | SoftTrain |
|--------------------|-------|-------|--------|-------|-----------|
| Preparation | 317.8 | 44.5 | 30.7 | 4.7 | 18.9 |
| Surrogate Training | 381.2 | 339.3 | 5523.1 | 313.4 | 949.5 |
| Total | 699.0 | 383.8 | 5553.8 | 318.1 | 968.4 |

method requires the most time as solving the gradient matching involves computation of second-order derivatives. Soft-Train method also spends more time compared to Craft-, GAN- and Train-ME because the soft-labels are used as the second objective.

In all the cases, the time cost of the proposed ME attacks is dominated by the cost of training the surrogate model. This heavily depends on the network topology, the number of iterations, and input size and vary from application to application, making it difficult to provide a comprehensive time complexity analysis.

A.11 Conditional-GAN architecture.

The detailed architecture of the conditional-GAN for GAN-ME attack is shown in Fig. 8.

```

Generator(
  (label_emb): Embedding(10, 512)
  (l1): Sequential(
    (0): Linear(in_features=1024, out_features=16384, bias=True)
  )
  (conv_blocks0): Sequential(
    (0): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
  (conv_blocks1): Sequential(
    (0): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): LeakyReLU(negative_slope=0.2, inplace=True)
  )
  (conv_blocks2): Sequential(
    (0): Conv2d(256, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): LeakyReLU(negative_slope=0.2, inplace=True)
    (3): Conv2d(128, 3, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (4): Tanh()
    (5): BatchNorm2d(3, eps=1e-05, momentum=0.1, affine=False, track_running_stats=True)
  )
)

```

Figure 8: Architecture detail of the c-GAN in GAN-ME.

B Model Inversion Attack Implementation

B.1 Model Inversion Attack Setting

Data privacy in SFL is evaluated using Mean Squared Error (MSE) between ground-truth images and reconstructed images in Model Inversion Attack (MIA). For MIA, we follow the same model-based attack methodology as in Vepakomma et al. [2020], Li et al. [2022]. The MIA flow is shown in Fig. 9. We assume the honest-but-curious attacker (this time, the server) has access to the 10K validation dataset of CIFAR-10. We use the L3 inversion model in Li et al. [2022] to perform MIA, and use the trained L3 inversion model to reconstruct the raw image from the intermediate activation sent by benign clients.

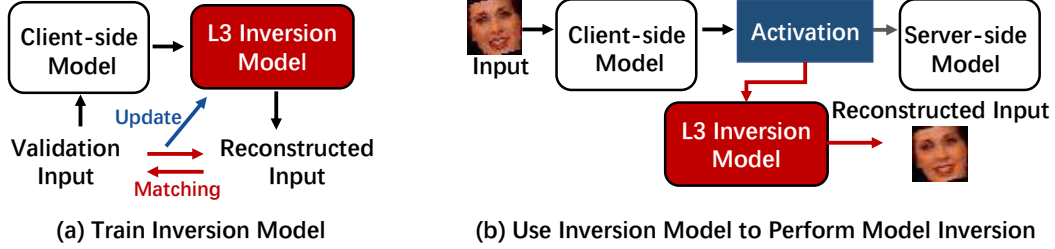


Figure 9: Details of model inversion attack using L3 inversion model and the available validation dataset, done by an honest-but-curious server. (a) Train the inversion model on the validation dataset. (b) Use the inversion model to invert intermediate activation sent by clients.