An Empirical Comparison of Gradient Leakage Attacks in Federated Learning - A Deeper Dive into CPL vs DLG

Saurab Sirpurkar
Georgia Institute of Technology
Atlanta, Georgia 30313
Email: ssaurab3@gatech.edu
Code can be found here

ABSTRACT

With its key feature being privacy, federated learning is a popular method for clients to train a global model using localized data. Despite this seemingly elevated sense of privacy, the clients are not fully out of risk. In this study we consider two attacks that demonstrated a privacy leakage in shared gradients from clients in federated learning. [1] shows how the gradients can be utilized to compute the training data of the clients for the first time. Similarly, [2] looks deeper into the training hyperparameters of the client as well as the attacker and provides insights into the effect of each of these on the attack cost and effect. In this paper, we look into the current state of these attacks on federated learning systems, mainly inspired by the experiments conducted in [2] and debate the current state of privacy that federated learning systems can expect.

1 INTRODUCTION

Federated learning has multiple advantages such as training efficiency, increased training data availability and participant privacy. But most of the privacy guarantees are based on assumptions such as "Privacy is enhanced by the ephemeral and focused nature of the Federated Learning updates" [3], or just based on that fact that it is better than sharing data for training. However, this is not entirely true.

There have been studies that observe attacks on privacy such as [4], [5] in federated settings. But in these studies, only a few properties of the training data are under threat of being revealed such as the majority background color of images. However, since 2020, attacks such as [1] threatened to reconstruct entire client datasets just using the shared gradients. More studies like [6] and [7] followed suit. Two attacks that are of particular interest to us in this paper [1] (DLG) and [2] (CPL).

The former attack [1] (referred to as DLG from here) stands as the first attack to reconstruct a client dataset using gradients in federated learning. They display the efficiency of such reconstruction in CV and NLP domains and achieve close to perfect reconstruction in most of the considered datasets and models. The main constraint with this attack is that the batch size has to be smaller than 8 and the in case of images, the resolution cannot exceed 64*64.

The latter attack [2] (referred to as CPL from here) takes up a similar formalism of iteratively changing a randomly initialised image until its gradient computed against the same model converges to the stolen gradient. However, one of the main differences of this attack is that it 'predicts' the label corresponding to the the stolen gradient through a theoretical argument that the sign of the gradient for the ground-truth class will be the opposite of other classes and usually also has the largest absolute value. Using this hypothesis, the authors directly take y' as $argmin_i(\nabla_i w_k(t))$ unlike in DLG where y' is also initialised randomly. Therefore, it has a headstart over it in terms of reconstruction speed. Moreover, the loss function for image reconstruction also differs from DLG with the addition of $\alpha||f(x^{\tau}_{rec}, w(t)) - y_{rec}||^2$ model loss term. The paper also looks into other experimental variations such as initialization techniques, larger batch sizes and image resolutions, performance when communication efficient gradients are used etc. Our goal is to compare CPL with DLG under these variations of architectures and hyperparameters.

2 DATASETS AND MODELS CONSIDERED

We narrow the scope of this study to the computer vision domain. Within CV, we mainly focus on the LFW dataset [8] shown in Fig. 1.

Moreover, we use the LeNet model in PyTorch as the main image classification model over both these datasets. The code for CPL and DLG are borrowed from here and are later modified for different values of variables. The final modified code can be found here here.

3 EXPERIMENTAL METHODS AND EVALUATION

We consider the overall performance of these attacks in multiple scenarios. This section explains the basic setup be-

hind these experiments.

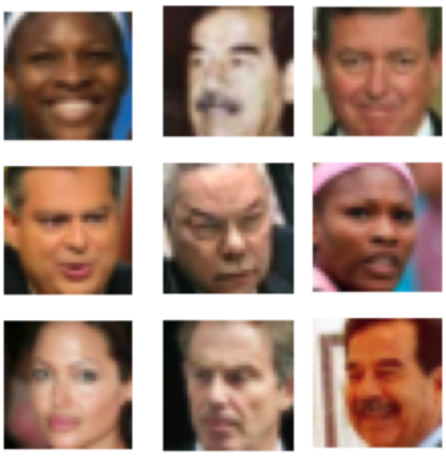

Fig. 1. A sample of images from the LFW dataset

3.1 Experimental Setup

3.1.1 Hardware

All of this experiment is performed on Google Colab with Hardware specifications as shown in the Fig. 2. A Tesla K80 GPU with 10GB GPU RAM was used along with a Intel Xeon(R) CPU with 12GB CPU RAM.

```
GPU 0: Tesla K80 (UUID: GPU-c3a6305e-95fe-9e80-fcbd-006d9a103f3e)
Sat Nov  6 06:26:00 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla K80           Off  | 00000000:00:04.0 Off |                    0 |
| N/A   49C    P8    31W / 149W |      3MiB / 11441MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
Model name:          Intel(R) Xeon(R) CPU @ 2.30GHz
CPU MHz:             2299.998
13G
Avail
32G

------------------------------------------------
            RESOURCE USAGE STATISTICS
------------------------------------------------

Gen RAM Free: 12.4 GB  | Proc size   : 1.2 GB
GPU RAM Free: 11438 MB | Used       :     3 MB | Util       :     0% | Total
```

Fig. 2. Specifications of the Hardware used

3.1.2 Attack Initialization

We experiment with all the initializations mentioned in [2] - random, fully-white, fully-black, level-1 CPL patterned and level-2 CPL patterned initial images. The difference between level-1 and level-2 patterns is that in the former, all four quadrants have the exact same pattern, where as in the latter, each half of every quadrant match exactly. The CPL attack paper mentions that the best reconstruction happens with such patterned initializations. We verify this by comparing the reconstruction of the attacks starting from each of these initializations. Moreover, we also verify if DLG can benefit from a patterned initialisation instead of a random one.

3.1.3 Batch Size

We experiment with different batch sizes such as 1, 5, 10, 15. The larger the batch size, the harder it is for attacks to succeed due to loss of specificity when averaging across the batch.

3.1.4 Iterations of Client Model Training

This is another parameter that affects the quality of reconstruction because the gradients become smaller in magnitude when the client's model is close to convergence. This is because, there are no major changes in the model weights that would be caused during backpropagation. For example, in an ideal hypothetical case when the gradient is zero in all dimensions, then there is no information we can gain from it about the training data. Therefore, the more rounds the client trains, the more attackers suffer in reconstruction quality.

3.1.5 Image Resolution

We consider with 32×32 images for most of the experiment. However, we also inspect the performance of CPL on 128×128 images and verify its convergence as per papers [2]'s claim.

3.2 Evaluation Metrics

The metrics considered in this experiment are focused on speed and accuracy of these attacks.

3.2.1 ASR

Gives the fraction of the training images that were successfully reconstructed using the stolen gradients under a specific experiment setup. This gives us the general rate of success that can be expected from each of these attacks.

3.2.2 MSE

MSE or the root mean square error is a metric to measure the deviation of the constructed image from the ground truth image. It can be calculated as : $\frac{1}{N}(x(i) - x_{rec}(i))^2$ where x is the ground truth image and x_{rec} is the reconstructed image.

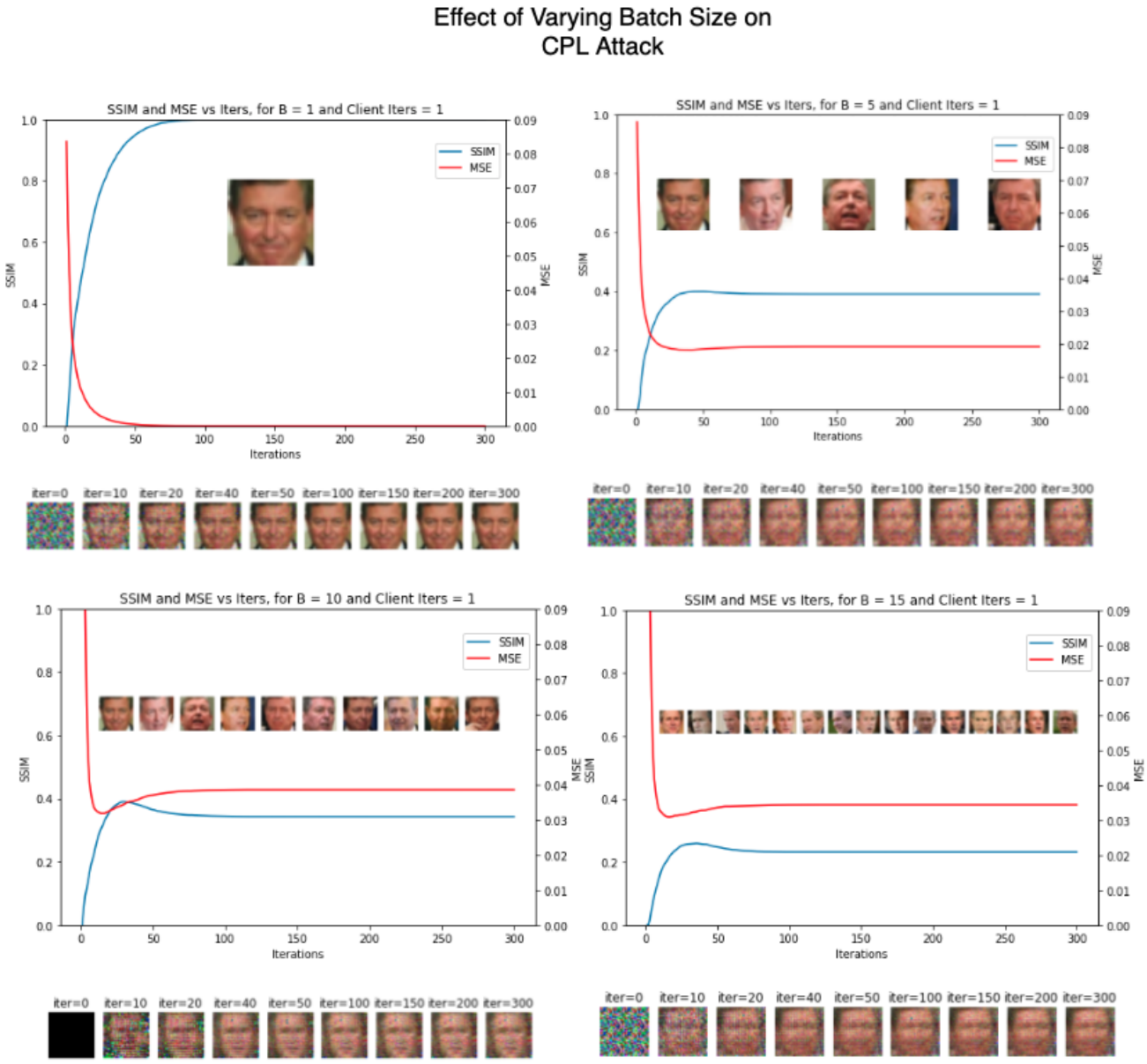

Fig. 3. Varying batch sizes in CPL attack with Single Client Iteration – the images inside the plot show the ground truth batch

3.2.3 SSIM

SSIM is an improved distance metric over MSE as it captures the structural similarity, thereby overcoming MSE's weakness of high sensitivity to contrast changes compared to base images. Its formula is as follows:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

3.2.4 Attack iterations and time

Another metric to evaluate a gradient leakage attack is the time is takes to generate the training data once it has the stolen gradient. This is proportional to the number of iterations it takes. This is important in characterizing attacks because in some not-so sensitive applications of federated learning, reconstructing older information may not be a problem. For example a financial institution that frequently releases its predictions and data publicly could tolerate a very slow federated learning attack. Therefore, this is an important metric to evaluate a gradient leakage attack.

4 EXPERIMENT RESULTS

For each of the mentioned variables such as the initialization, batch size, client iterations and image resolution, a thorough empirical study has been conducted documenting

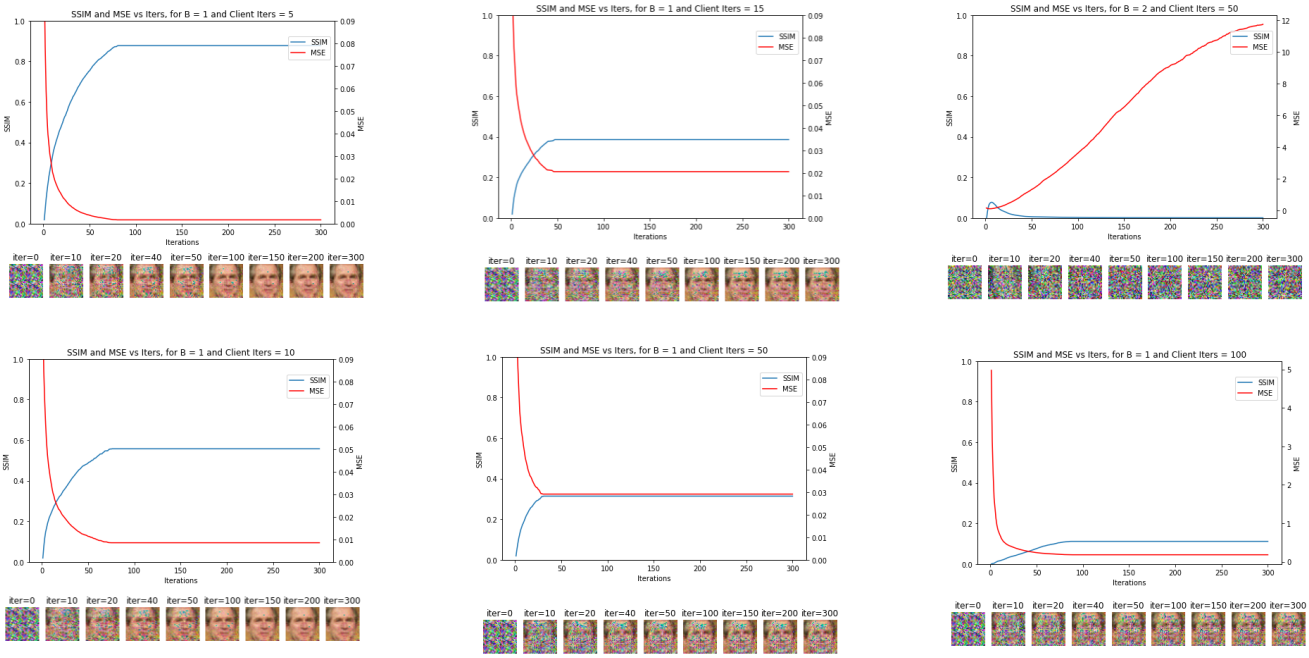

Fig. 4. Varying client training iterations in CPL attack

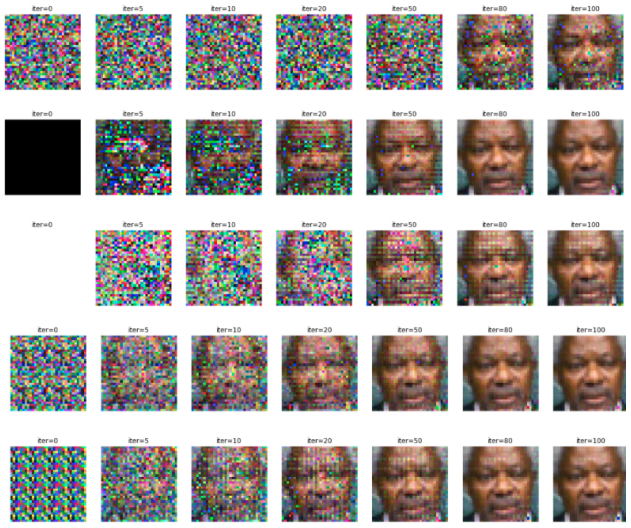

Fig. 5. Performance of CPL using different initializations from 0 to 100 iterations

the results in this section. In the next session, we go deeper on the analysis of these results.

4.1 Attack Initialization

The figures Fig. 5 and Fig. 6 inspect the reconstruction performance of CPL on two different gradients by using each of the given initializations for the dummy images. In order, we first see totally random initialization, fully-black initialization, fully-white initialization, level-1 patterned initialization, level-2 patterned initialization. Next, Fig. 7 shows DLG attack starting from the random initialization and the level-2 patterned initialization for two different classes. This is a new experiment that is not covered in either of the papers, but similar results are available at AI-Privacy vLab of DiSL, Georgia Tech. This experiment reconfirms that the proposed patterned initialization also makes DLG converge faster. We will discuss more about this in the next section.

4.2 Batch Size

A varying batch size was considered form the set {1,5,10,15} to examine the effect of batch size on CPL and DLG attacks. The Fig. 3 shows the variations of CPL Attack efficacy (measured in terms of SSIM, MSE vs No. of Iters) as the batch size varies. The reconstruction progress through attack iterations is shown below the plot and the ground-truth batch is inside the plot. Also, Fig. 8 shows a similar plot for DLG. We observe that it fails at B=10 unlike CPL that continues to perform reasonably well until B=15.

4.3 Iterations of Client Model Training

Fig. 4 shows the effect of client model-training-iterations on the reconstruction quality of the training dataset by the attacker. Here, we considered iterations from the set: {5,10,15,50,100} for B=1 and B=2. For spatial efficiency, only a salient plot is shown for B=2. As expected, for a constant batch size, as the number of iterations at the client increases, the quality of reconstruction by the attacker goes down.

Fig. 6. Performance of CPL using different initializations from 0 to 100 iterations

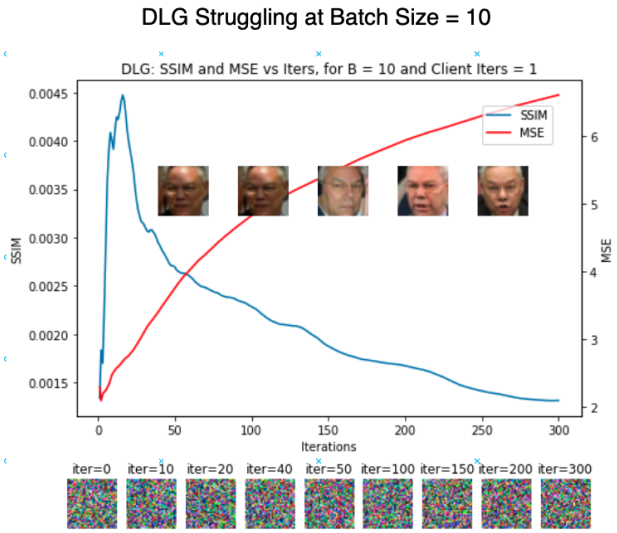

Fig. 8. Performance of DLG at B=10

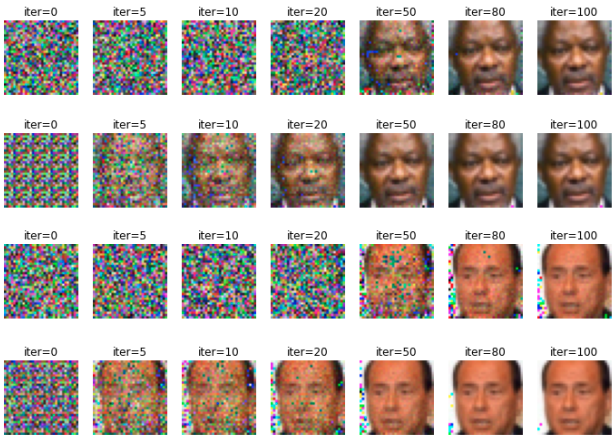

Fig. 7. Performance of DLG using random and patterned initialization from 0 to 100 iterations for 2 different classes

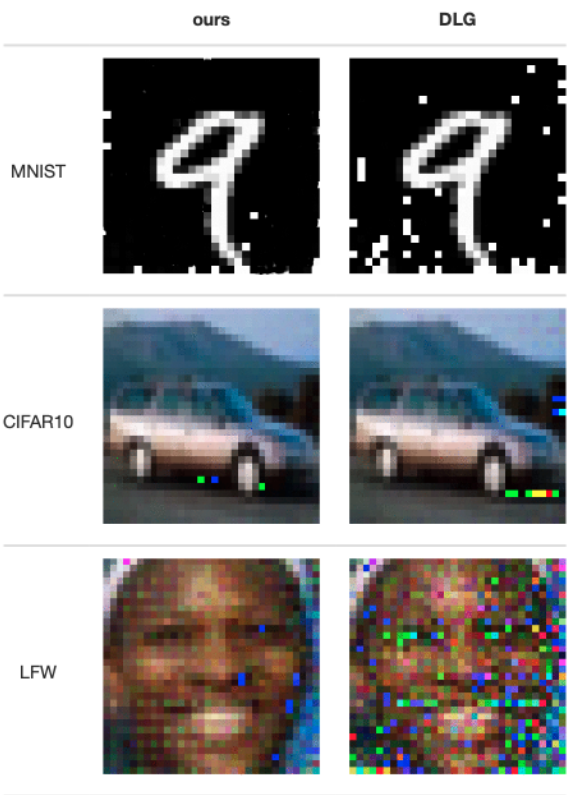

Fig. 9. An AI-Privacy vLab Sample Comparison

4.4 Image Resolution

According to [1], DLG attack works only on images upto 64×64. However, Fig. 10 shows a successful reconstruction of a 128×128 training images using CPL. This shows that CPL has a capacity to reconstruct images with significantly higher number of pixels. We discuss the possible reason behind this in the next section.

5 ANALYSIS AND DISCUSSION

5.1 AI-Privacy vLab

The AI-Privacy vLab has some results of experiments similar to the ones conducted in this paper. For example, the Fig. 9 shows a sample comparison for a certain number of iterations between CPL and DLG. This web application is useful to quickly grasp the main ideas behind these attacks and compare them on a surface level. Supplementing the understanding one could gain from this app, our current analysis looks into a deeper level by studying how these attacks behave when other variables change.

Fig. 10. Using CPL with patterned initialization to reconstruct a training image with 128×128 resolution

5.2 CPL vs DLG Analysis

We have seen that CPL primarily differs from DLG in three ways - it uses a predicted label from the gradient instead of a random one, it uses the model loss function along with attacker gradient loss as the main loss and it comes in multiple initialization variants. Throughout this experiment, since CPL came out on top in all the categories, it is conclusive that both these tweaks are working towards a faster and/or a better attack. For example, **4.1** discusses the effect of different patterned initializations on CPLs speed of convergence. Here, we see that regular DLG that uses a random initialization converges much slower than the patterned CPL variants.

Next, to compare across batch sizes, observe Fig. 3 vs Fig. 8. It is clear that CPL proves to be a better attack, given that all images in the batch come from the same class. This can again be attributed to the three differences between CPL and DLG. It is likely that the 'predicted' label is helping in initial convergence, and later the modified loss function is helping in final convergence towards a reasonable reconstruction. Finally, observe that a majority of these experiments use level-2 patterned initialization which could be an additional reason behind faster convergence and better results.

Next, we look at the effect of client model iterations count. From Fig. 4, we can see that CPL maintains a reasonable reconstruction somewhere until 25 iterations. Observe that LeNet converges fairly quickly and this is still a decent improvement over DLG which is known to suffer when client iterations increase as the gradient vanishes. A curious observation is the top-right subplot in Fig. 4 in which initially we expect the attack to move towards convergence, but it suddenly starts to get worse. One possible explanation behind this could be that since the gradient is small, after the initial convergence push by the predicted $y' = arg\,min_i(\nabla_i w_k(t))$, the model loss term $\alpha||f(x^{\tau}_{rec}, w(t)) - y_{rec}||^2$ starts to dominate the smaller gradient difference, therefore sidetracking towards a random reconstruction that perhaps minimizes this model loss for the assumed class. An extension to this study would be to fiddle with the hyperparameter α's values and adjust it to improve the attack when client iterations increase.

Finally, we see the evidence that CPL can work for higher resolutions in Fig. 10, where reasonable reconstruction is achieved within 220 iterations. The DLG authors claims that this is not possible using their attack. The reason why CPL potentially has an upper-hand in this case is again likely that the predicted label $y' = arg\,min_i(\nabla_i w_k(t))$ is acting as a guiding force in the first few iterations unlike the random labels in DLG. Once the image moves in the hyperspace in the initial right direction, it is likely that it can have more success in converging. This can be seen in the image where the main features of the person are visible within the first 30 iterations showing that it is plausible that this can be attributed to the effect of correctly guessed y', perhaps together with the modified loss function.

5.3 Datasets

Although this paper mainly considers LFW dataset, the transferability of these finding is high across datasets. For example, the AI-Privacy vLab shows the same reconstruction attack for CIFAR and Fashion-MNIST and it performs similarly on both these datasets.

6 CONCLUSION

This paper examined two state-of-the-art gradient leakage attacks CPL and DLG and compared the two across a broad range of experimental conditions. CPL came out on top in almost all of these experiments. Later, it discussed the possible reasons behind this.

Overall, attacks like DLG continue to show that Federated Learning can no longer be considered as a perfectly safe way to train models. CPL pushes this notion further by experimenting with a variety of realistic scenarios such as higher batch sizes and resolutions, and paired with its original contributions of loss function, initialization and label prediction, it clearly establishes the imminent threat of such attacks on Federated Learning systems. Any organization or individual currently participating in such training networks should be aware of these advancements and rethink if Federated Learning continues to be suitable for them.

References

[1] Zhu, L., and Han, S., 2020. "Deep leakage from gradients". In *Federated learning*. Springer, pp. 17–31.

[2] Wei, W., Liu, L., Loper, M., Chow, K.-H., Gursoy, M. E., Truex, S., and Wu, Y., 2020. "A framework for evaluating client privacy leakages in federated learning". In European Symposium on Research in Computer Security, Springer, pp. 545–566.

[3] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečnỳ, J., Mazzocchi, S., McMahan, H. B., et al., 2019. "Towards federated learning at scale: System design". *arXiv preprint arXiv:1902.01046*.

[4] Phong, L. T., Aono, Y., Hayashi, T., Wang, L., and Moriai, S., 2017. "Privacy-preserving deep learning: Revisited and enhanced". In International Conference on Applications and Techniques in Information Security, Springer, pp. 100–110.

[5] Melis, L., Song, C., De Cristofaro, E., and Shmatikov, V., 2019. "Exploiting unintended feature leakage in collaborative learning". In 2019 IEEE Symposium on Security and Privacy (SP), IEEE, pp. 691–706.

[6] Geiping, J., Bauermeister, H., Dröge, H., and Moeller, M., 2020. "Inverting gradients–how easy is it to break privacy in federated learning?". *arXiv preprint arXiv:2003.14053*.

[7] Yin, H., Mallya, A., Vahdat, A., Alvarez, J. M., Kautz, J., and Molchanov, P., 2021. "See through gradients: Image batch recovery via gradinversion". In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16337–16346.

[8] Huang, G. B., Mattar, M., Berg, T., and Learned-Miller, E., 2008. "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments". In Workshop on faces in'Real-Life'Images: detection, alignment, and recognition.