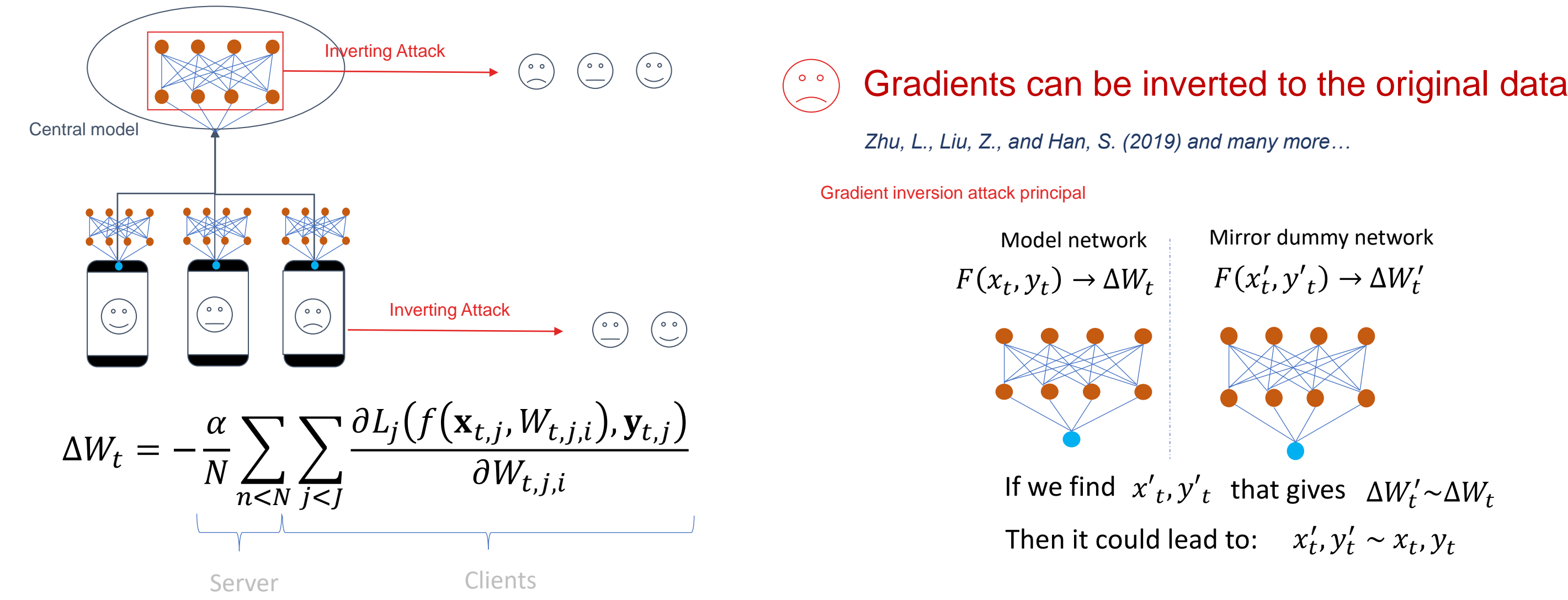


Mixing Gradients in Neural Networks as a Strategy to Enhance Privacy in Federated Learning

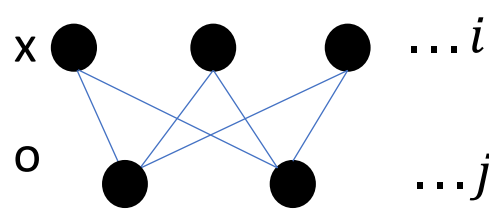
Shaltiel Eloul, Fran Silavong, Sanket Kamthe, Antonios Georgiadis, and Sean J. Moran
CTO, JPMorgan Chase

Federated learning reduces the risk of information leakage, but remains vulnerable to attack. We show that well-mixed gradients provide numerical resistance to gradient inversion in neural networks. For example, we can enhance mixing gradients in a batch by choosing an appropriate loss function and drawing identical labels, and we support this with an approximate solution of batch inversion for linear layers. These simple architecture choices show no degradation to classification performance as opposed to noise perturbation defense. To accurately assess data recovery, we propose to use a variation distance metric for information leakage in images, derived from total variation. In contrast to Mean Squared Error or Structural Similarity Index metrics, it provides a continuous metric for information recovery. Finally, our empirical results of information recovery from various inversion attacks and training performance supports our defense strategies. These simple architecture choices found to be also useful for practical size of convolutional neural networks but depends on their size. We hope this work will trigger further defense studies using gradient mixing, towards achieving a trustful federation policy.

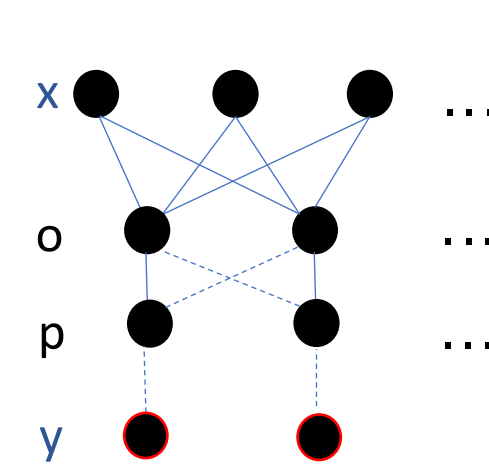
Federated learning controls distributed learning from various resources, where the main goal is not sharing or re-locating the data. However it is shown that optimization attacks can be highly successful in inverting back to the data:



Exercise - Revisiting the vulnerable dense layer in typical classification (Direct inversion)

$$o_j = \sum_i w_{ij} x_i + b_j$$


In a classification problem and explored attacks benchmarks use soft-max followed by cross entropy:

$$p_k = \frac{e^{o_k}}{\sum_j e^{o_j}}$$
$$l(p, y) = -\sum_k y_k \log p_k$$


We discuss here the last dense layer of a network, but any input of previous layer, can be propagated back so any dense layer before x can be found.

The set equations for change of loss that is shared in FL:

$$\frac{\partial l}{\partial w_{i,j=k}} = (p_j - y_j) x_i$$
$$\frac{\partial l}{\partial b_{j=k}} = p_j - y_j$$

For inversion of data this can be done directly, with no optimizer:

$$\frac{\partial l}{\partial w_{i,j=k}} / \frac{\partial l}{\partial b_{j=k}} = x_i$$

This is well known.

Exercise for Batch - Direct inversion in a Batch

What about a batch, then we only share the sum of changes:

$$\frac{\partial l^B}{\partial w_{i,j=k}} = \frac{1}{B} \sum_{m \in [1, B]} (p_j^m - y_j^m) x_i^m$$

B-Batch size, m-index in batch:

$$\frac{\partial l^B}{\partial b_{j=k}} = \frac{1}{B} \sum_{m \in [1, B]} p_j^m - y_j^m$$

This is supposedly make it difficult to invert without optimization. However in unique batch this is not the case:

$$p_j - y_j = p_j - 0, \quad j \neq c(\text{not class})$$
$$p_j - y_j = p_j - 1, \quad j = c(\text{true class})$$
$$\frac{\partial l^B}{\partial b_{j=k}} \approx \frac{((p_j^m - 1) x_i^{m(j=c)} + (p_j^m - y_j^m) \sum_{j \neq c} x_i^{m(j \neq c)})}{((p_j^m - 1) + (B-1)(p_j^m - y_j^m))} = x_i^{m(j=c)}$$

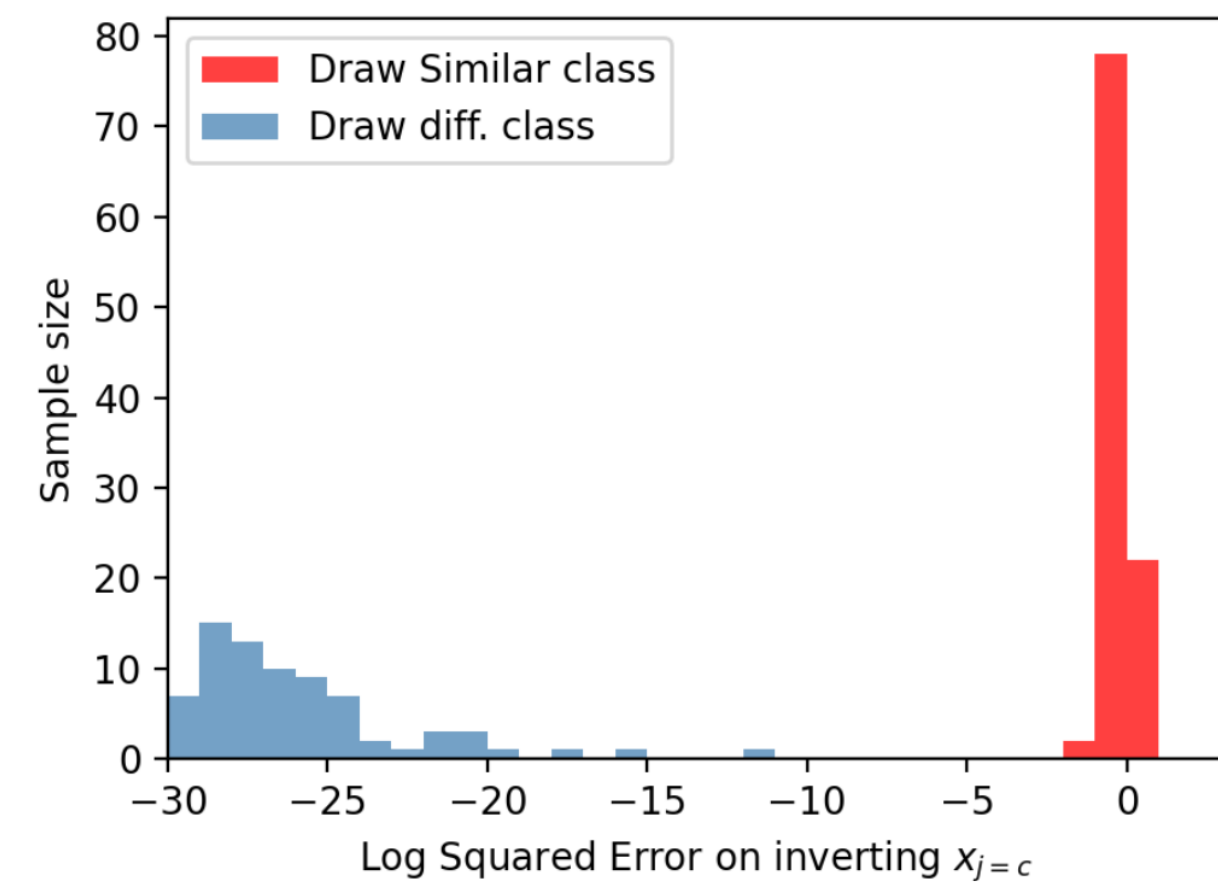
We can neglect the terms due to the approximation and obtain x directly for the whole batch:

$$E(p_j(o_j)) = p_j(E(o_j)) + Err(O^2)$$
$$\langle p_j \rangle \propto \frac{1}{C} \text{ Inversely proportional to no. of Classes}$$

However this does not work if we have similar labels in the same batch. We get a mixed gradients:

$$\approx \frac{\sum_i ((p_j^m - 1) x_i^{m(j=c)} - x_i^{m(j \neq c)})}{\sum_i ((p_j^m - 1) - 1)} = \sum_i x_i^{m(j=c)}$$

Error on accurate recovery of data in direct inversion for the full batch using the direct inversion:

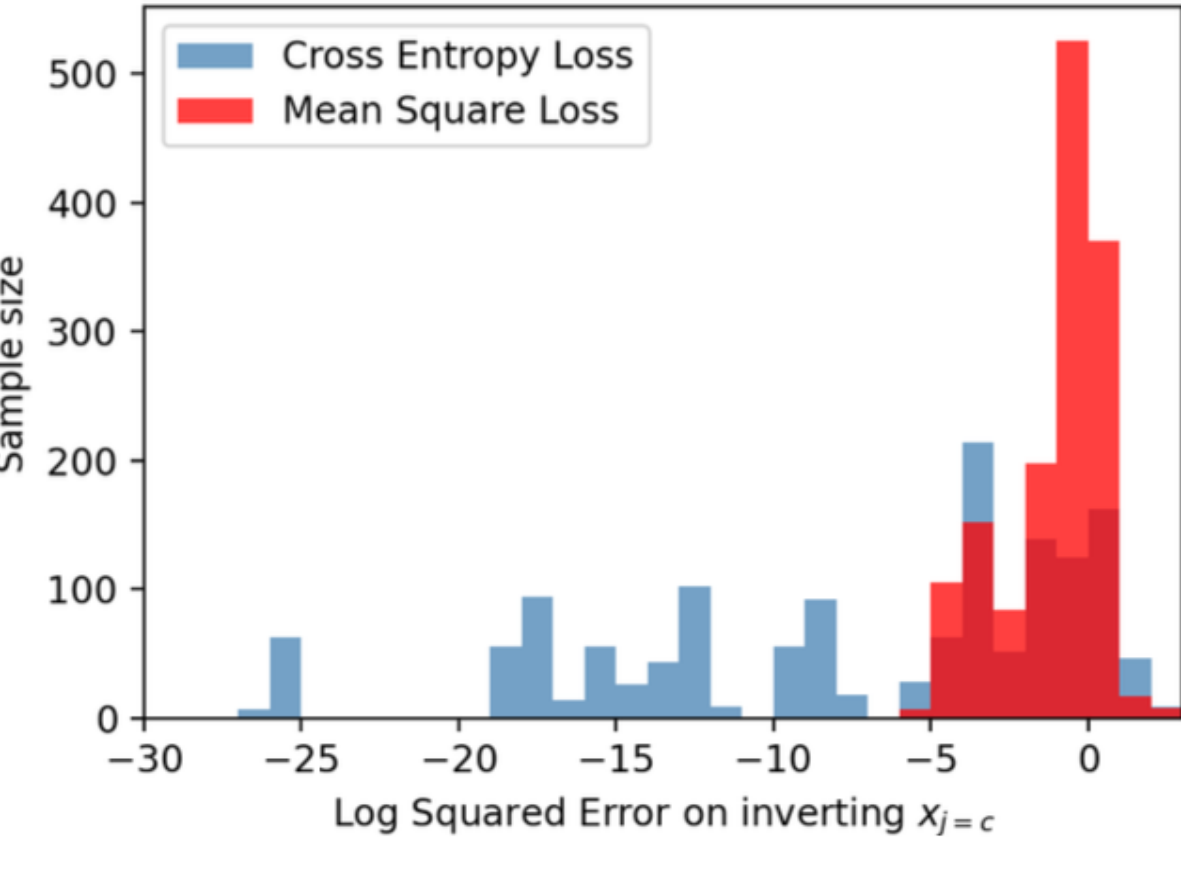


Direct inversion is successful and accurate for different classes in a batch, without any optimization, unless there are similar labels, in which the error is in the order of the input x.

With similar claims we can replace the loss function to mix gradients in a batch. Specifically in Mean Square Error loss, we obtain the gradients:

$$\frac{\partial l_2}{\partial w_{i,j=k}} = \frac{\partial l_2}{\partial o_k} \frac{\partial o_k}{\partial w_{i,j=k}} = -2(o_j - y_j) x_i$$

$$\frac{\partial l_2}{\partial b_{j=k}} = -2(o_j - y_j)$$

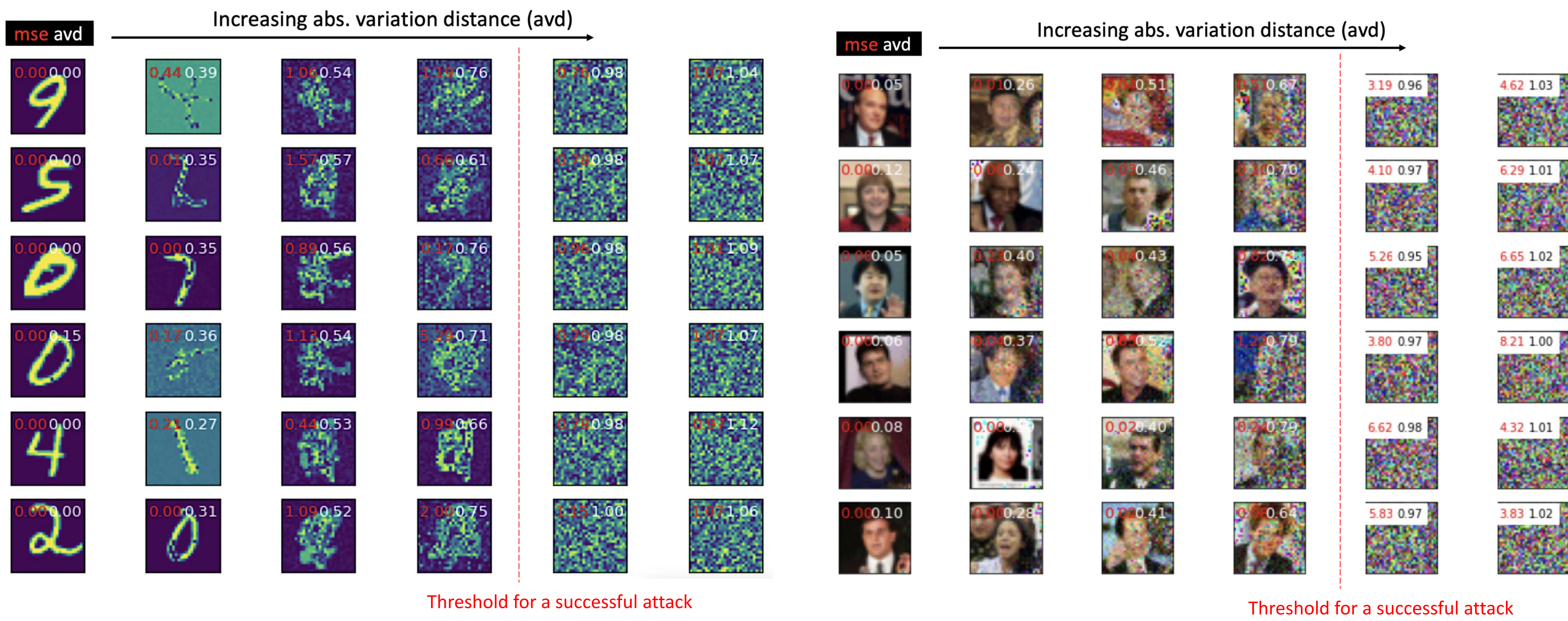


In this case, the gradients do not depend on p, and therefore are not decay to zero, and proportional to the output y. The mixing is natural and does not depend on the class.

Experimental Setup on Optimization attacks

We investigate the success rate of attacks on the configuration of NN by using several optimization attacks, and we look at revealing minimal information from a picture. In order to evaluate a successful attack, we can't use mean Square error, or other pixel wise metrics presented in previous studies.

We hence introduce a different metric, that seems to work very well on such benchmark data: $AVD(v_{source}^{source}, v_{target}^{target}) = ||(|\nabla v_{x,y}^{source}| - |\nabla v_{x,y}^{target}|)||$



Optimisation of inversion attacks used for the study

$$g^{l^2}(x'_t, y'_t) = \min ||\Delta W'_t - \Delta W_t||$$

$$g^{ang}(x'_t, y'_t) = \min 1 - \frac{\langle \Delta W'_t, \Delta W_t \rangle}{||\Delta W'_t|| \cdot ||\Delta W_t||}$$

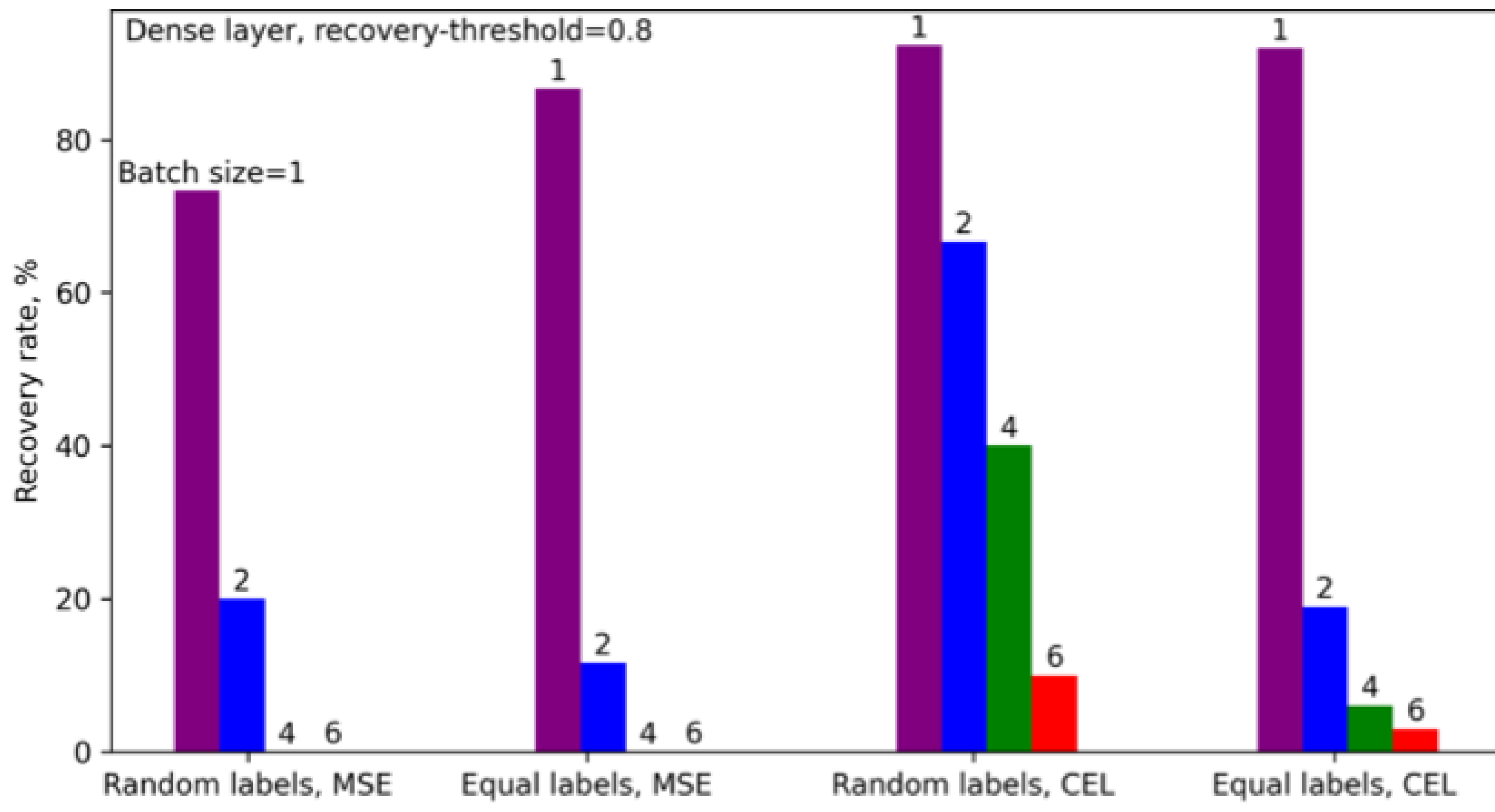
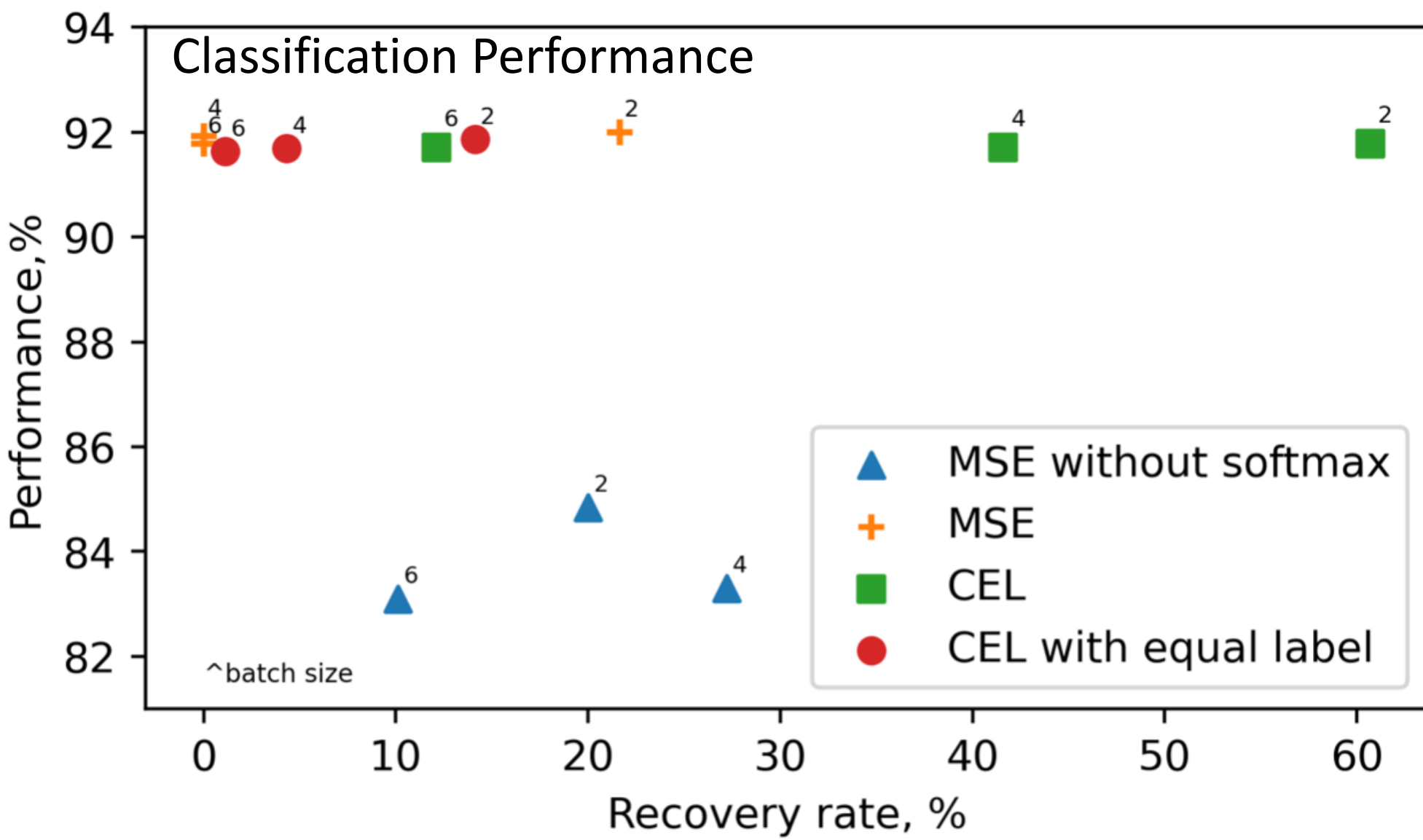
With priors/constraints:

- Total variation.
- Orthogonality of images in a batch.
- Determination of y labels, using gradients.

Benchmark datasets: MNIST, LFW

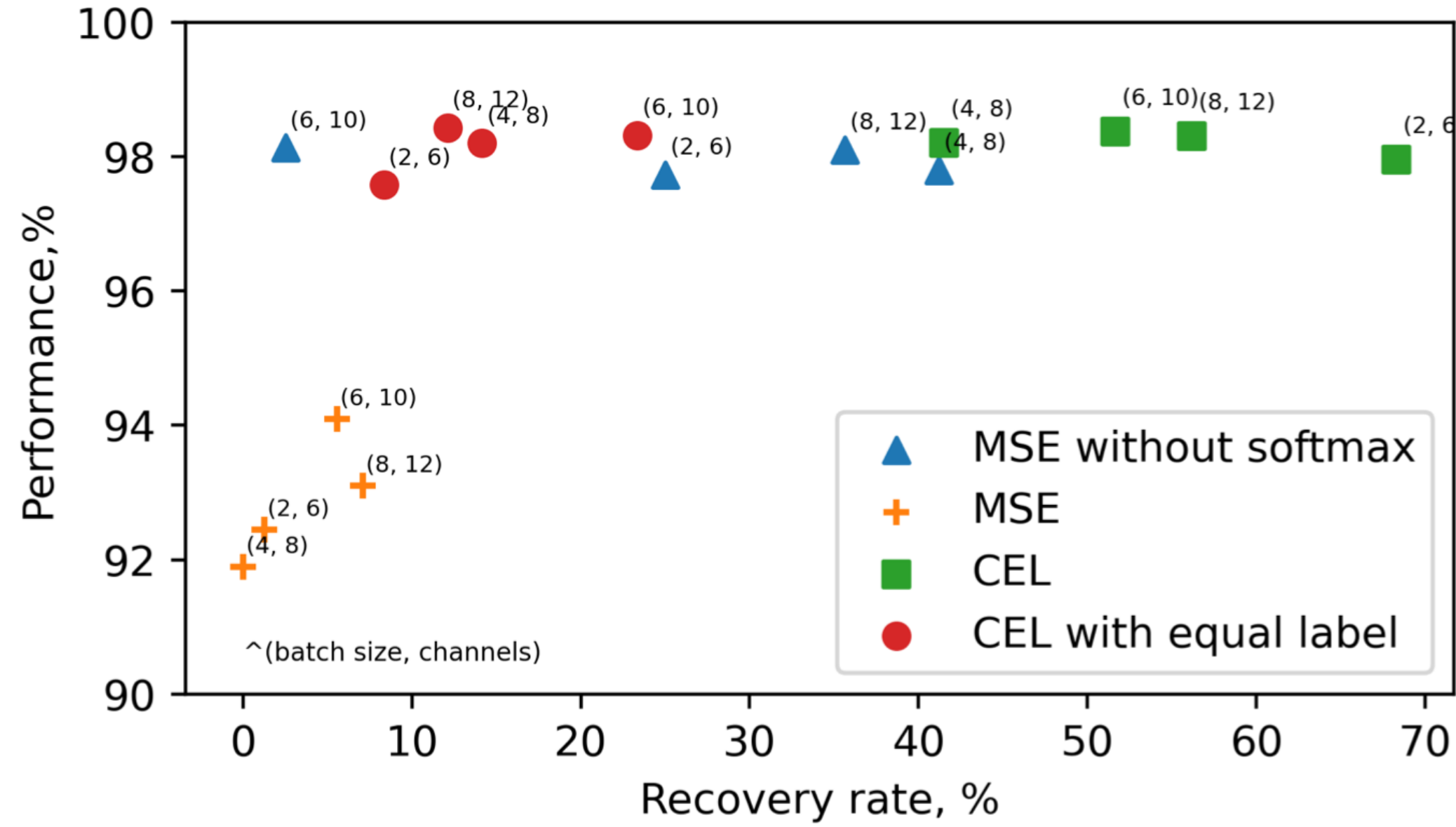
Experimental Results – Dense Layer – Privacy and Performance

Choosing equal labels in batch and mean square error (MSE) instead of Cross entropy loss (CEL) helps in mixing and reduce the recovery rate in attacks test. Classification performance is also remains high for some configurations with very low recovery rates.



Experimental Results – CNN - LeNET– Privacy and Performance

In CNN, when large number of channels are used, the strategies are less relevant because data can be inverted only with the convolutional layers. However using mixing strategies still helpful and increase security. Additional noise can be added on gradients to reduce recovery rates to zero in large number of channels.



Conclusions

The choice of loss function and the drawing of equal labels in a batch results in mixing of the gradients in practical neural networks architectures. In fact, without mixing gradients, it is possible to recover directly all batch vectors due to the de-mixing nature of cross entropy loss function. Our suggested strategies for mixing gradients maintain network performance in certain setups, which is in contrast to common methods that apply noise to the gradients. Additionally, in practice, one could combine the mixed gradients strategies further with noise or other defense methods for better privacy. Finally, we have shown that an absolute variation distance (AVD) metric is able to measure the relative information recovered by gradient inversion attacks. The metric, which is derived from total variation, can distinguish information from noise for datasets that have sparse information such as in the MNIST dataset and will be explored further in future studies. We hope that this work prompts the development of new strategies towards achieving more trustful federated learning platforms. Further work will also study the effect of more complex architectures and larger models which are more challenging area of privacy preserving in distributed learning.