PRECODE - A Generic Model Extension to Prevent Deep Gradient Leakage – Supplementary Material –

Daniel Scheliga[1] Patrick Mäder[1,2]
Marco Seeland[1]

[1]Technische Universität Ilmenau, [2]Friedrich Schiller Universität Jena, Germany
{daniel.scheliga,patrick.maeder,marco.seeland}@tu-ilmenau.de

1. Overview

In sections 2 and 3 we specify the loss functions used for the reconstruction of images in the conducted experiments as well as the applied metrics for evaluating the attacks. Furthermore, the hyperparameters, positioning and quantity of PRECODE modules are discussed in section 4. Finally we provide the accuracy curves that describe the models training progress (Fig. 1-3) and more exemplary reconstruction results for the CIFAR-10, CFIAR-100 and Medical MNIST datasets (Fig. 4-9).

2. Attack Loss Functions

The Client Privacy Leakage (CPL) attack [3] extends the objective of minimizing the euclidean distance of the dummy gradient and the observed gradient as in DLG/iDLG by a label-based regularization term to stabilize the optimization.

$$\arg\min_{x'} ||\nabla\mathcal{L}_\theta(F(x),y) - \nabla\mathcal{L}_\theta(F(x'),y)||^2 + \alpha||F(x'),y||^2, \tag{1}$$

where α tunes the impact of the regularization term on the optimization.

To perform the IGA attack described in [1] the dummy image x' is optimized for:

$$\arg\min_{x'} 1 - \frac{\nabla\mathcal{L}_\theta(F(x),y)\cdot\nabla\mathcal{L}_\theta(F(x'),y)}{||\nabla\mathcal{L}_\theta(F(x),y)||||\nabla\mathcal{L}_\theta(F(x'),y)||} + \alpha\mathrm{TV}(x'). \tag{2}$$

The total variation of the reconstructed image $\mathrm{TV}(x')$ is added as a simple image prior, where α determines its weight during optimization. The output labels y can be easily determined from the gradients as discussed before [4].

3. Metrics

In our experiments MSE measures the pixelwise squared error as:

$$\mathrm{MSE}(x,x') = \frac{1}{n_p}\sum_{i=1}^{n_p}(x_i - x_i')^2, \tag{3}$$

where n_p is the number of pixels in the images x and x'. Lower MSE values indicate a higher image similarity. PSNR, which is commonly used to assess lossy compression schemes in video and imaging, is defined as:

$$\mathrm{PSNR}(x,x') = 10\log_{10}\left(\frac{\max(x)^2}{\mathrm{MSE}(x,x')}\right) \tag{4}$$

The higher the PSNR value the better the better the reconstruction. SSIM uses a perception based model to measure the structural similarity between two images [2]:

$$\mathrm{SSIM}(x,x') = \frac{(2\mu_x\mu_{x'}+c_1)(2\sigma_{xx'}+c_2)}{(\mu_x^2+\mu_{x'}^2+c_1)(\sigma_x^2+\sigma_{x'}^2+c_2)}, \tag{5}$$

where μ_x, σ_x and $\mu_{x'}$, $\sigma_{x'}$ are the mean and variance values for x and x' respectively. $\sigma_{xx'}$ denotes the covariance of x and x', $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$. $k_1 = 0.01$ and $k_2 = 0.03$ are set by default and L is the dynamic range of pixel values. SSIM values closer to 1 indicate a higher similarity between images.

4. PRECODE VB Hyperparameters and Positioning

We investigated the influence of the VB hyperparameters $k \in [64, 128, 256, 512]$ and $\beta \in [10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}]$ as well as the position and quantity of PRECODE modules in the model on the training process and quality of reconstructions.

If multiple bottlenecks are added to the model, the loss function is adjusted accordingly:

$$\mathcal{L}(\hat{y},y) = \mathcal{L}_F(\hat{y},y) + \sum_{i=0}^{n}\mathcal{L}_{P_i}, \tag{6}$$

where

$$\mathcal{L}_{P_i} = \beta_i D_{KL}(\mathcal{N}(\mu_{B_i}, \sigma_{B_i}), \mathcal{N}(0, 1)), \tag{7}$$

and β_i tunes the weight of the ith PRECODE module P_i with bottleneck layer B_i on the overall loss function.

We found that the choice of k had no notable impact on the training process, the final model performance and the quality of reconstructions. If the choice for β was too high (*i.e.* $\beta = 10^{-1}$), the time for convergence increased and model performance decreased. For the other β values there was no further impact observed. Again we found no notable impact on the quality of reconstructions. We placed PRECODE after four different feature extracting layers in a MLP with four hidden layers and found that a placement too close to the input resulted in a slightly increased quality of image reconstructions. Nevertheless the content of the training data was still unrecognizable and model performance was not impacted. Placing more than one PRECODE module into the network resulted in a decrease of model performance but did not impact the quality of reconstructions.

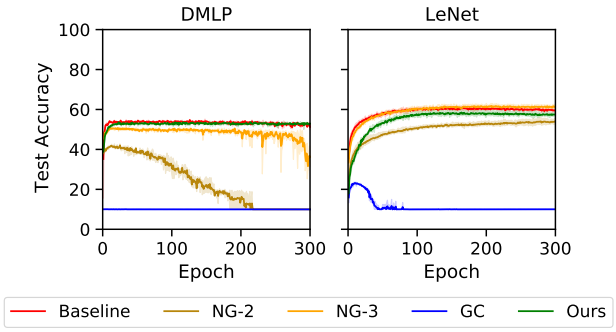
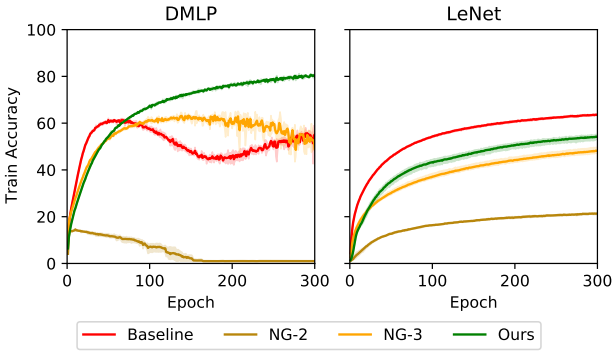
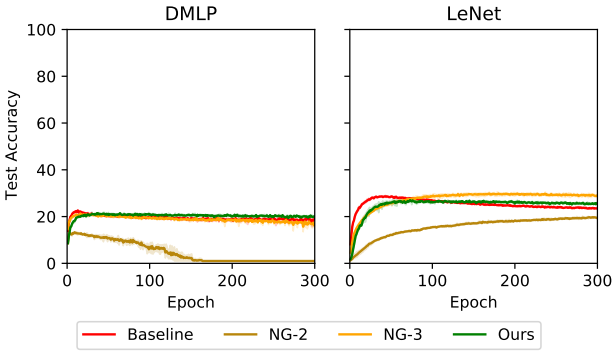


Figure 1: Test accuracy on the CIFAR-10 dataset. Line colors define the baseline model and defense mechanisms.

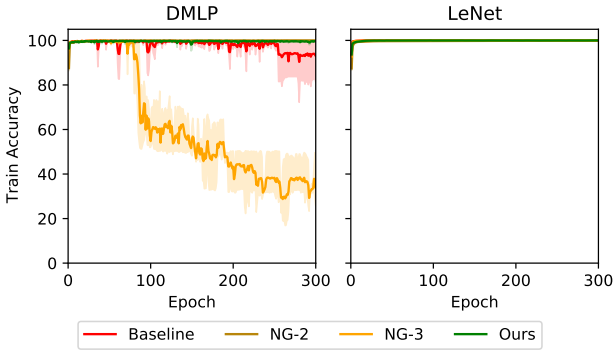


(a)

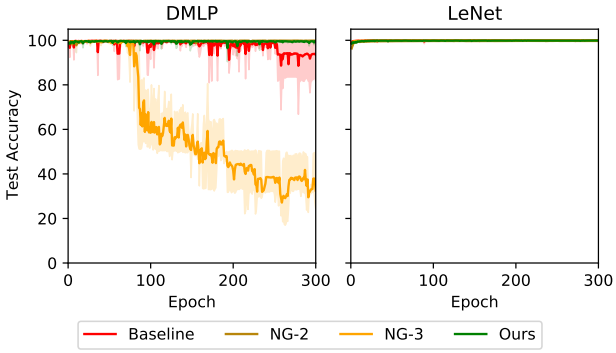


(b)

Figure 2: Train accuracy (a) and test accuracy (b) on the CIFAR-100 dataset. Line colors define the baseline model and defense mechanisms.



(a)



(b)

Figure 3: Train accuracy (a) and test accuracy (b) on the Medical MNIST dataset. Line colors define the baseline model and defense mechanisms.

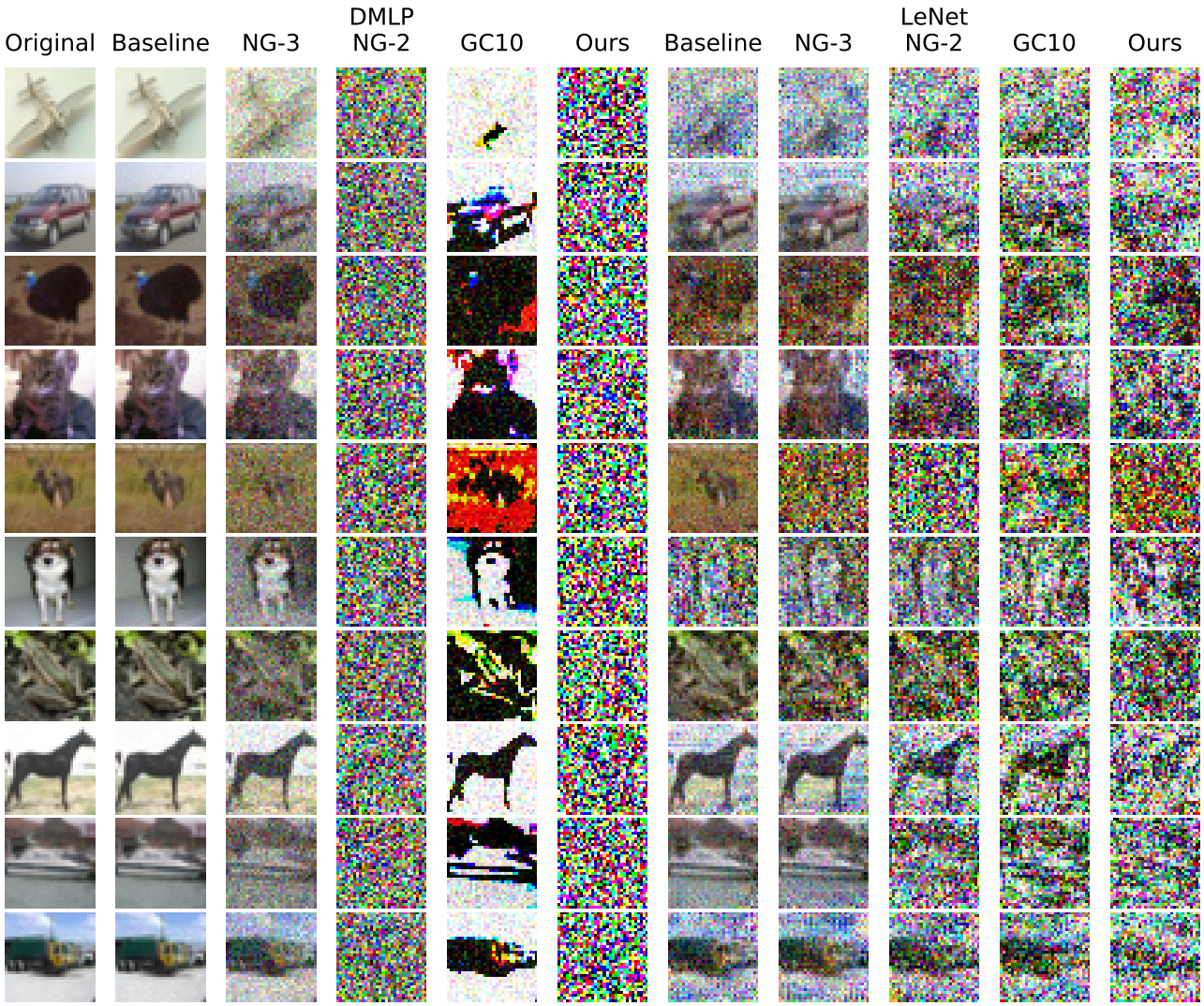


Figure 4: Exemplary reconstruction results for all 10 classes of the CIFAR-10 dataset for the baseline DMLP and LeNet models and different defense mechanisms.

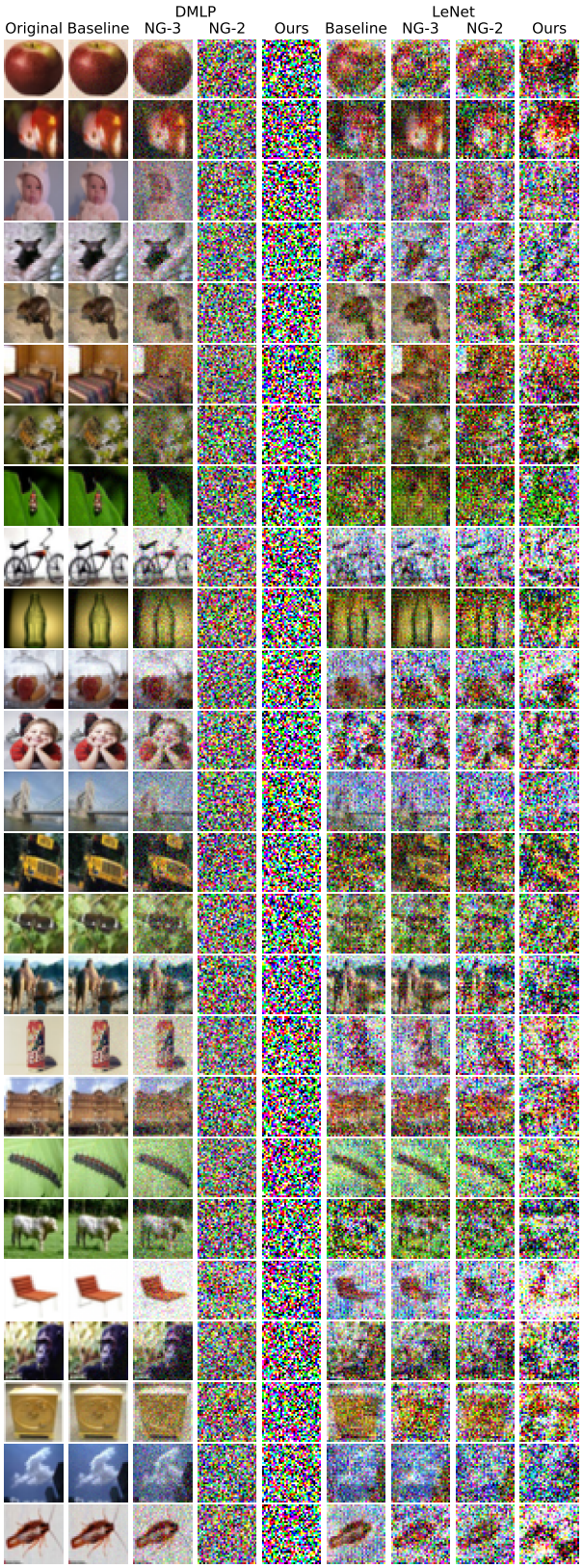


Figure 5: Exemplary reconstruction results for classes 1-25 of the CIFAR-100 dataset for the baseline DMLP and LeNet models and different defense mechanisms.

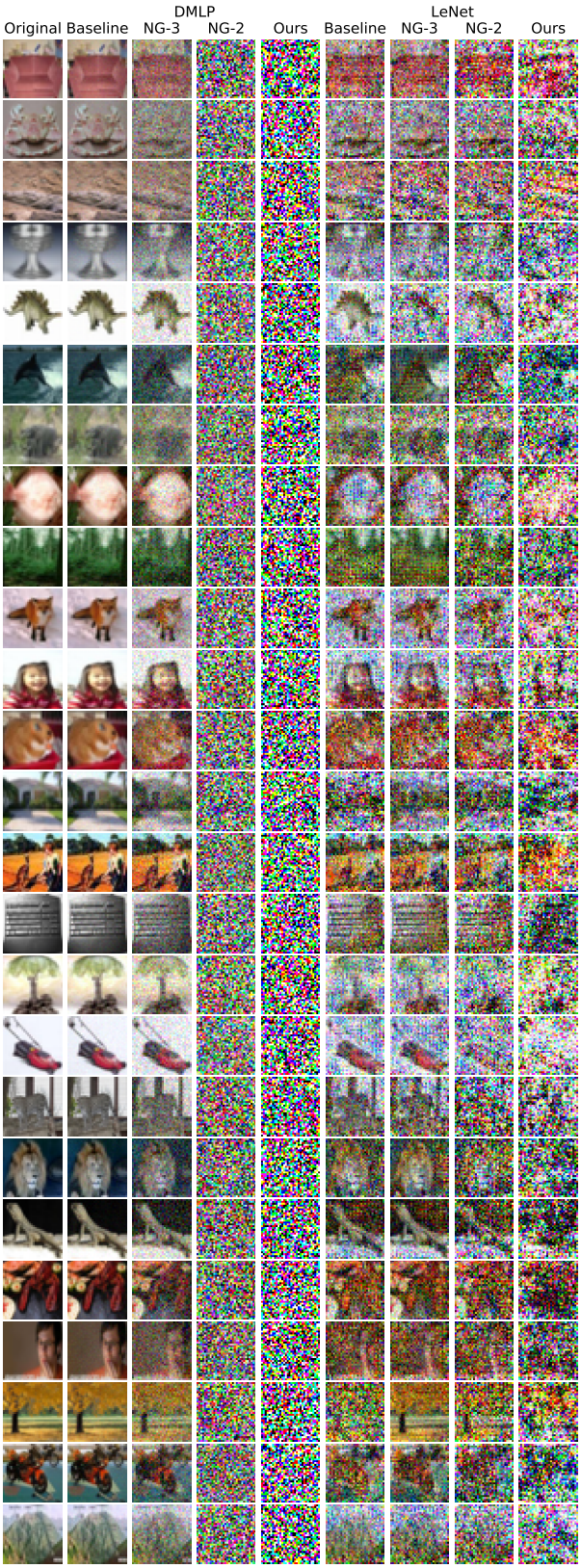


Figure 6: Exemplary reconstruction results for classes 26-50 of the CIFAR-100 dataset for the baseline DMLP and LeNet models and different defense mechanisms.

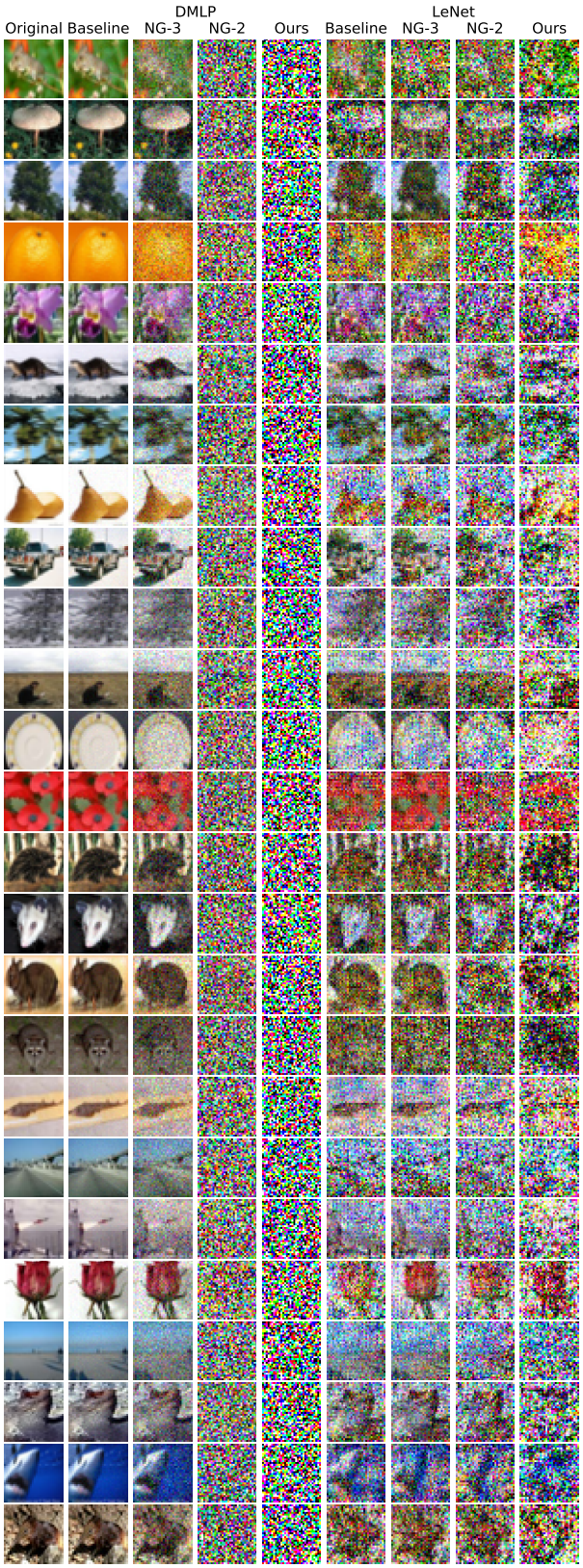


Figure 7: Exemplary reconstruction results for classes 51-75 of the CIFAR-100 dataset for the baseline DMLP and LeNet models and different defense mechanisms.

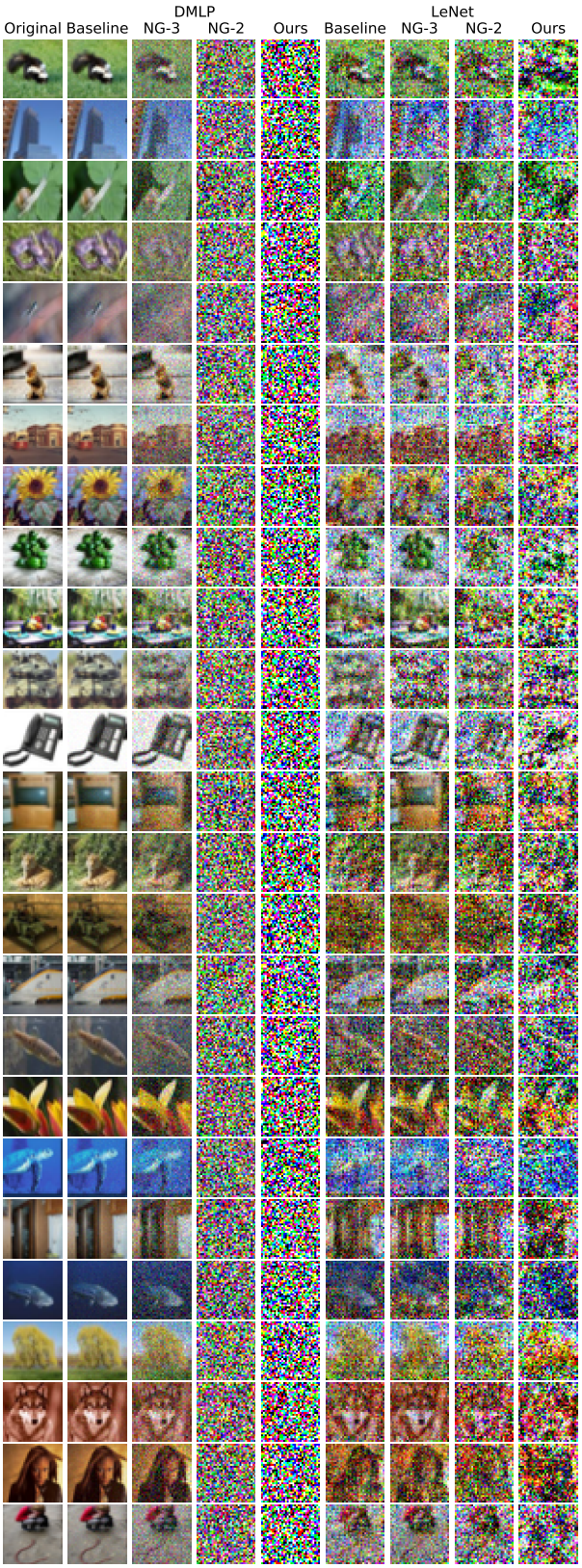


Figure 8: Exemplary reconstruction results for classes 76-100 of the CIFAR-100 dataset for the baseline DMLP and LeNet models and different defense mechanisms.

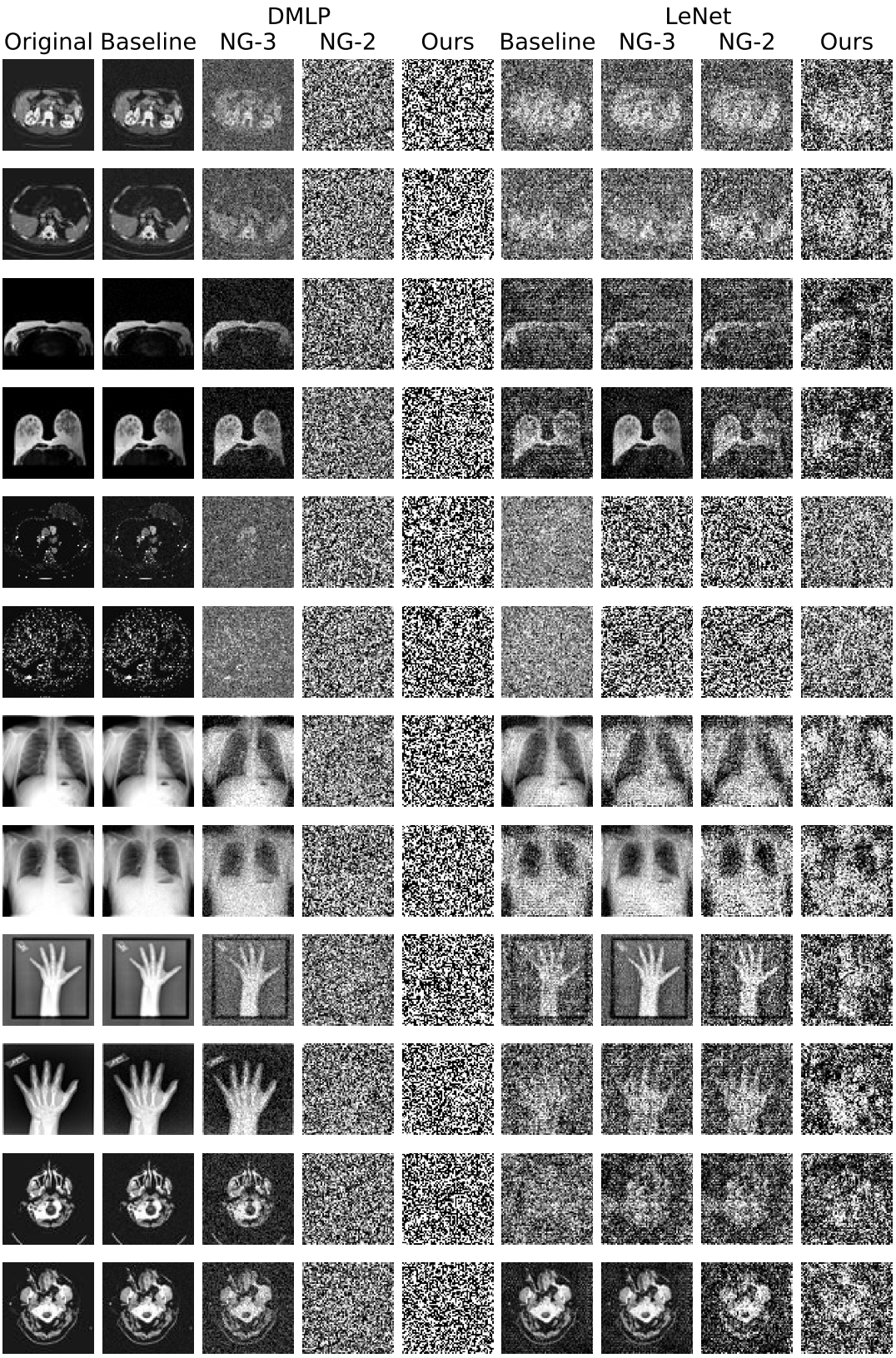


Figure 9: Exemplary reconstruction results for all 6 classes of the Medical MNIST dataset for the baseline DMLP and LeNet models and different defense mechanisms.

References

[1] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients - how easy is it to break privacy in federated learning? In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16937–16947. Curran Associates, Inc., 2020.

[2] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[3] Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. A framework for evaluating gradient leakage attacks in federated learning. *arXiv preprint arXiv:2004.10397*, 2020.

[4] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020.