A. Additional Reconstruction Samples

Due to page limit, we only include the reconstruction results under the Soteria [44] defense in our main paper (Figure 5) for additional visualization samples on the ImageNet dataset. Here we present the full results under all 4 considered defenses (i.e., additive noise [44, 56] with $\sigma = 0.1$, gradient clipping [14, 48] with $S = 4$, gradient spasification [56] with a pruning rate of 90%, and Soteria [44] with a pruning rate of 80%) in Figure 10. We observe that our method is able to reconstruct high-quality images from gradients in all these considered cases regardless of the type of defense.

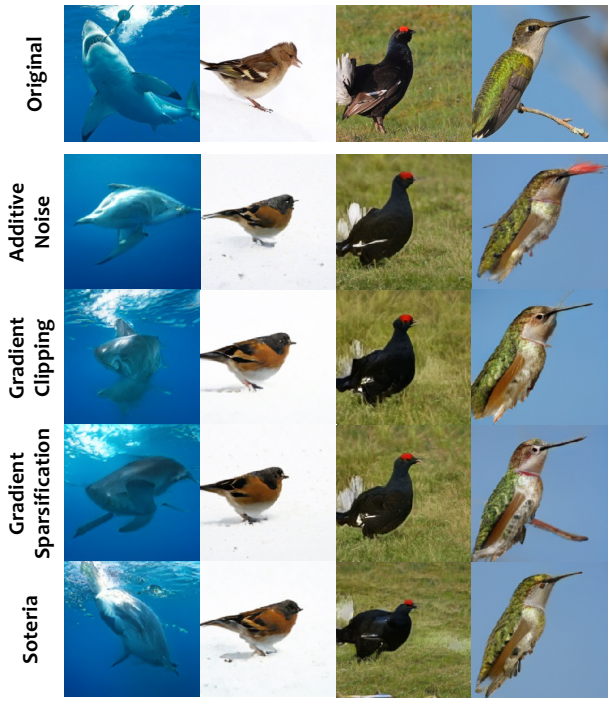

Figure 10. Reconstruction results under various defenses on the ImageNet dataset: (first row) original images and (the rest of rows) their reconstructions by GGL under various defenses.

B. Implementation Details

Optimization Configuration. We use the following configuration for the explored optimizers: (1) *Adam*: initial learning rate $lr = 0.1$, $\beta_1 = 0.9$, $\beta_2 = 0.999$. On the CelebA dataset, we use a step learning rate decay at step 937, 1562, and 2189, by a factor of $\gamma = 0.1$. On the ImageNet dataset, the learning rate is linearly warmed-up from 0 during the first 125 iterations and gradually reduced to 0 in the last 625 iterations using cosine decay; (2) *BO*: We use the *TurBO-1* algorithm [10] with 256 initial points, batch size = 10, lower bound = -2, upper bound = 2, and automatic relevance determination (ARD) kernel for the Gaussian process; and (3) *CMA-ES*: we use random initialization with batch size = 50. We set $\lambda = 0.1$ for experiments on the CelebA dataset. On the ImageNet dataset, for algorithms that do not innately support bound constraints, we apply the *tanh* function to achieve the bound.

GAN Configuration. For the CelebA dataset, we train a DCGAN [40] with a latent dimension of 128 with its detailed structure presented in Figure 11. Specifically, we use the Wasserstein distance with the loss weight set to 10 for the gradient penalty [17]. The GAN model is trained for 100 epochs using Adam optimizer with a learning rate of 0.0001 and a batch size of 64. For the ImageNet dataset, we use a pre-trained BigGAN [6] with a latent dimension of 128 and output image size of 256×256. The output image is further rescaled to 224×224 for computing the FL task.

Type	Kernel	Stride	Output
FC			8192
BN1D			8192
DeConv2D	2×2	2×2	256
BN2D			256
DeConv2D	2×2	2×2	128
BN2D			128
DeConv2D	2×2	2×2	3

(a) Generator

Type	Kernel	Stride	Output
Conv2D	3×3	2×2	128
Conv2D	3×3	2×2	256
Conv2D	3×3	2×2	512
FC			1

(b) Discriminator

Figure 11. GAN structure for the CelebA dataset.

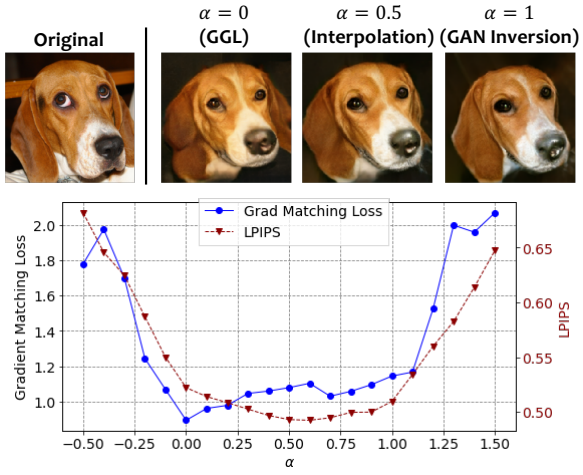

Figure 12. Comparison of image reconstructed by our method and GAN inversion.

C. Loss Landscape Analysis

Comparison with GAN Inversion. In our attack, we consider the private image to be unknown and the adversary attempts to reconstruct the image from the shared gradient information using a pre-trained GAN. However, such reconstruction is constrained by the generator's fitting ability. GAN inversion technique which inverts a given image to the GAN's latent space can serve as a means for testing the upper bound of the image quality reconstructed from GAN. To evaluate, we compare the reconstructed image from gradients using our method and the inverted image using GAN inversion technique [25]. To compare the information provided by gradient information with the information provided by the original image, we further visualize the gradient matching loss and the LPIPS loss in the GAN latent space. Specifically, we plot the loss functions by interpolating between the latent vectors found by the proposed GGL ($\mathbf{z}_1$) and GAN inversion ($\mathbf{z}_2$): $\mathbf{z}(\alpha) = (1-\alpha)\mathbf{z}_1 + \alpha\mathbf{z}_2$. From the results presented in Figure 12 we observe that (1) the latent vector found by our method does yield the lowest gradient matching loss on this line; (2) compared to the gradient information, the information provided by the original image can better guide the optimization process in the GAN latent space: the latent vector found by GAN inversion produces a better image quality (lower LPIPS) than the solution found by our method; and (3) the latent vector with the lowest gradient match loss doesn't result in the best image quality/similarity (measured by LPIPS).

Different Defenses. We next analyze how each defense mechanism affects the loss landscape. We extend the visualization to a 2D surface by adding a second random direction vector $\boldsymbol{\eta}$ (normalized according to $\mathbf{z}_2 - \mathbf{z}_1$): $\mathbf{z}(\alpha, \beta) = \mathbf{z}_1 + \alpha(\mathbf{z}_2 - \mathbf{z}_1) + \beta\boldsymbol{\eta}$. Figure 13 shows the visualized loss surface under different defense settings. We can see that additive noise and gradient sparsification do not have much impact on the geometric landscape of the gradient matching loss, whereas gradient clipping and Soteria [44] clearly deform the gradient matching loss surface, rendering it hard for the adversary to find a good reconstruction under such defenses. However, by applying the adaptive transformation at the adversary's side, such deformation can be greatly mitigated and thereby enables the adversary to reconstruct high-quality images even with the presence of these defenses.

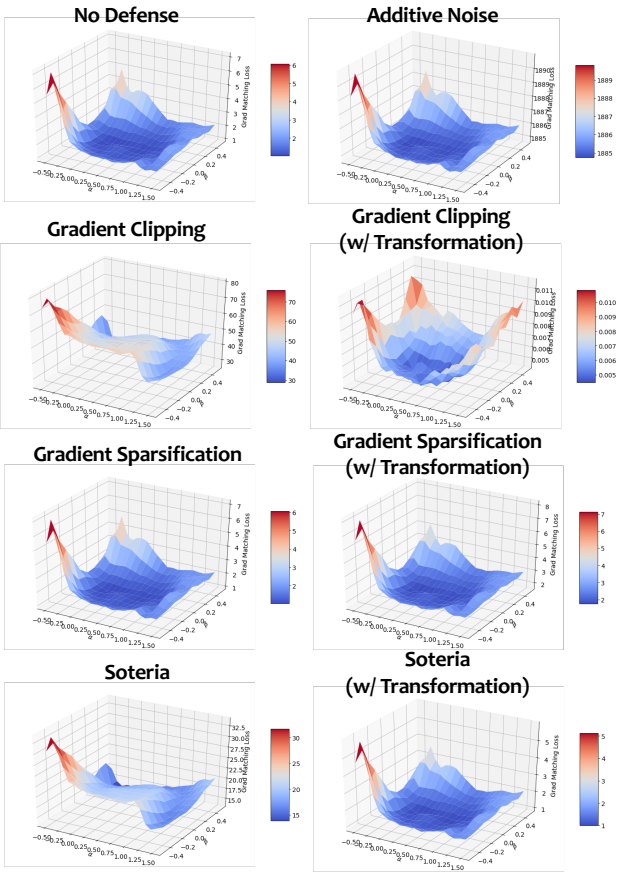

Figure 13. Visualization of observed loss landscapes under various defense settings. The bottom 3 rows compare the loss surface with (right) and without (left) applying adaptive transformation at the adversary's side.

D. Larger Batch Sizes or Multiple Local Steps

Recovering high-resolution batch data with multiple local steps remains a major challenge in this line of research. Most existing studies [13, 56] only work on small images (32×32px) for batch size > 1. Currently, the only study that accounts for local steps > 1 is IG [13], but it only works on a single ImageNet image. The only study that can work on batched full-size ImageNet images (224×224px) is GI [51], which supports up to 48 images with local step $= 1$. However, it can only reveal limited information from partial images of the batch, and it assumes that the BatchNorm (BN) statistics (mean and std.) of the target batch is jointly provided with the gradients and only works for specially pre-trained large ResNet-50 model (larger model provides more gradient information).

Differently, we seek to investigate the privacy leakage under various defense strategies. We show that even with batch size $= 1$ and local step size $= 1$, existing methods still failed to reconstruct the input under defenses, while our method can reveal a good amount of visual information.

To investigate the generalizability of GGL, we conducted additional experiments on batched ImageNet images (224×224px) and with multiple local steps, with the results presented in Figure 14 and Figure 15, respectively. We can see that GGL can still restore a decent amount of visual information under these settings. The proposed GGL can be

further strengthened with additional prior information (e.g., BN statistics).

Figure 14. Image reconstruction with batch size = 4: (1st row) original images, (2nd row) reconstructions by GGL w/o defense, and (3rd row) reconstructions by GGL w/ Soteria [44] defense.

Figure 15. Reconstruction by GGL with multiple local steps.

Figure 16. Reconstruction of *in-the-wild* images: (1st row) images from *Google Images* and (2nd row) their reconstructions by GGL.

E. Recovering In-the-wild Data

We target the practical scenario where the attacker can utilize all public-accessible data as prior information to launch the attack. Thus we chose to use CelebA and ImageNet for evaluation as they are all Internet-based datasets and are easy to access as an attacker. We also used the disjoint dataset so that the images used for testing haven't been used for GAN training. To investigate the performance of GGL under the scenario where the testing image is not from the GAN training distribution, we conducted additional experiments to recover *in-the-wild* images (i.e., arbitrary images from the search results in Google Images with appropriate cropping/resizing). From the results in Figure 16, we can see that GGL can still reveal a reasonable amount of visual information even if the testing images are not from the GAN training distribution.