Consistent Low-Rank Dequantization for Gradient Inversion under Low-Bit Quantization

Seoklea Cho and Jinu Gong
Department of Applied AI
Hansung University
Seoul, Korea
2171269@hansung.ac.kr, Jinugong@hansung.kr

***Abstract*—In federated learning, sparsification and low-bit quantization are known to mitigate the effectiveness of gradient inversion attacks by removing information during the model update process. However, these evaluations consider attacks that simply restore the received quantized or sparsified updates. A more sophisticated adversary that exploits prior knowledge can still successfully carry out an attack. We propose consistent low-rank dequantization (CLR), a preprocessing method for the Cocktail Party Attack under low-bit quantization. The proposed algorithm, CLR, utilizes only the received codes, scales, and known batch size, and applies row-normalized low-rank approximation and projection onto relaxed quantization intervals in an alternating fashion. Experiments on disjoint ImageNet batches show that CLR improves feature recovery over standard dequantization and row normalization alone. Also, we confirm that performance improvements at the feature level lead to enhanced image quality. In addition, we conduct experiments regarding bit width and rank. These findings suggest that standard dequantization alone is insufficient for assessing the privacy protection provided by quantization and that evaluations should also consider structure-aware decoding.**

***Index Terms*—Federated learning, uplink quantization, gradient inversion, low-rank recovery, consistent reconstruction.**

I. Introduction

Model updates can expose private training data through gradient inversion attacks that reconstruct the trained images by matching gradients [1], [2]. More recently, the Cocktail Party Attack (CPA) [3] interprets a fully connected (FC) update as a mixture of per-image features and separates the features using independent component analysis (ICA). In bandwidth-constrained wireless federated learning environments, clients often quantize their uplink updates to the level of a few bits per entry. In general, sparsification and quantization can degrade the performance of gradient inversion attacks by removing information from model updates. Previous studies have examined compression as a defense [4], while others have shown that private information can survive gradient obfuscation [5].

However, quantization appears to provide protection only because the attacker naively dequantizes the received values. In fact, what a quantized value provides is a feasible set, and we dequantize it via round-to-nearest, either for simplicity or because no additional information is available. If the attacker has information that enables more effective dequantization, this conclusion may change. Therefore, evaluations based solely on standard dequantization may overestimate the privacy protection provided by quantization.

Since structure-aware attacks inherently depend on the assumed the training structure, we adopt the setting of the Cocktail Party Attack (CPA) [3], a recently proposed gradient inversion attack that recovers multiple images from a batch, and extend it to quantized updates. When the client trains repeatedly on the same batch, the update of the first fully connected (FC) layer is a linear mixture of fixed input features, and its rank is bounded by the batch size. Based on this property, we propose consistent low-rank dequantization (CLR). CLR approximates the received update by a low-rank matrix and clips each entry to the quantization interval of its received code, since the low-rank approximation alone is generally inconsistent with the transmitted codes. The two steps are applied alternately, and the reconstructed update is fed to CPA.

Our contributions are summarized as follows. First, we demonstrate that the rank of fully connected (FC) layer updates is constrained by the batch size when a client performs training using a single batch, and we analyze how quantization distorts this structure. Second, we propose CLR, a technique that alternates between low-rank approximation and clipping tailored to quantization intervals. Third, using the ImageNet dataset, we show that CLR significantly improves feature recovery performance compared to standard dequantization methods; we also examine the impact of each component, as well as the sensitivity to bit width and the assumed rank.

II. Background and Related Work

A. Gradient inversion and CPA

Optimization-based gradient inversion searches for synthetic inputs whose gradients match an observed update [1], [2]. Recovering many images jointly becomes difficult as the batch grows [6]. CPA [3] exploits the FC gradient structure to formulate feature recovery as a source-separation problem.

For an FC layer $\mathbf{z} = \mathbf{W}\mathbf{h} + \mathbf{b}$, a sample contributes $\boldsymbol{\delta}\mathbf{h}^\top$ to the weight gradient, where $\boldsymbol{\delta} = \partial\mathcal{L}/\partial\mathbf{z}$. Summing over B samples gives a linear mixture of B input features. CPA whitens a B-dimensional principal representation of the

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) in 2026 (Grant No. RS-2026-25587247, Project Title: Research on Principled Unlearning in Larger Language Models: Substantial Forgetting via Information Residual Bounds and Structural Fine-tuning, and its Multimodal Expansion).

observed update and optimizes an unmixing matrix. Recovered features are then mapped to images by feature inversion through the public extractor, with a total-variation (TV) prior.

Both subspace estimation and source separation affect recovery. Linear unmixing cannot recover feature components absent from the input subspace. However, accurate subspace estimation alone does not guarantee successful separation or faithful image reconstruction.

B. Compression and privacy leakage

Compression changes the gradient or update available to an attacker. Mixed quantization has been proposed as a defense against gradient inversion [4]. Other work shows that quantization, sparsification, and perturbation can provide a misleading impression of privacy protection [5].

Reconstruction methods have also been developed to account for degraded observations. Generative gradient leakage [7] uses a learned generative prior, while compression-aware gradient attacks incorporate compression into the reconstruction procedure [8]. Our approach instead uses the transmitted quantization intervals and the FC update structure to improve the numerical update supplied to CPA.

C. Consistent reconstruction

Consistent reconstruction seeks a signal whose quantized values match the observations. This idea was studied for oversampled signals and quantized frame expansions [9], [10]. In the latter setting, an estimate can be refined by alternately projecting it onto the signal subspace and the set defined by the quantization intervals.

Moshtaghpour et al. [11] studied quantization consistency together with structural constraints, including low rank, for dithered random linear measurements. Here, the observations are directly quantized FC updates with a separate scale for each row. We normalize the rows by these scales, then alternate rank truncation and interval projection before passing the reconstructed update to CPA.

III. Observation Model

A. Client update and rank bound

The attacker observes quantized update without the aggregation and knows the public model, batch size B, and training configuration, but not the images, labels, features, or unquantized update. The client trains FC layers for T SGD steps on the same batch, with a frozen extractor, deterministic preprocessing, dropout disabled, and no momentum or weight decay.

Let $\mathbf{H} = [\phi(x_1), \dots, \phi(x_B)]^\top \in \mathbb{R}^{B\times d}$ contain the fixed FC1 inputs. At step t, the gradient is $\mathbf{A}_t\mathbf{H}$, with batch averaging absorbed into $\mathbf{A}_t \in \mathbb{R}^{m\times B}$. In exact arithmetic,

$$\mathbf{U} = \mathbf{W}^{(T)} - \mathbf{W}^{(0)} = -\eta\Big(\sum_{t=0}^{T-1} \mathbf{A}_t\Big)\mathbf{H}, \tag{1}$$

where $\operatorname{rank}(\mathbf{U}) \le B$. For different batches or changing input features, the update instead produce $\sum_t \mathbf{A}_t\mathbf{H}_t$, and the bound B need not hold.

B. Quantization and relaxed consistency

For b bits, symmetric per-channel quantization can be described as

$$s_i = \|\mathbf{U}_{i,:}\|_\infty / Q_b, \qquad q_{ij} = \operatorname{round}(U_{ij}/s_i), \tag{2}$$

with $Q_b = 2^{b-1} - 1$ and $q_{ij} \in \{-Q_b, \dots, Q_b\}$. Per-tensor quantization uses a common matrix-maximum scale. The packet contains codes $\mathbf{Q}$ and FP32 scales $\mathbf{s}$, and standard decoding returns $U^{\mathrm{ctr}}_{ij} = s_i q_{ij}$. Then, the reconstruction is constrained by the closed interval as

$$\mathcal{C}(\mathbf{Q}, \mathbf{s}) = \{\mathbf{U}' : \ell_{ij} \le U'_{ij}/s_i \le u_{ij}\}, \tag{3}$$

where $\ell_{ij} = \max(q_{ij} - \frac{1}{2}, -Q_b)$ and $u_{ij} = \min(q_{ij} + \frac{1}{2}, Q_b)$. Note that this box is a relaxation, since it treats the received scales $\mathbf{s}$ as fixed constants. In reality, each s_i is determined by the maximum magnitude in row i.

IV. Consistent Low-Rank Dequantization

CPA estimates a feature subspace from the observed update before separating individual sources. Quantization can distort the estimated subspace, for example, by suppressing weak source components. Standard dequantization returns the scaled integer codes, although the received codes and scales permit alternative update values within the quantization intervals.

In the fixed-batch, frozen-extractor setting, the pre-quantization update has rank at most the batch size in exact arithmetic. We propose consistent low-rank dequantization to combine the rank bound with the quantization-interval constraints. The decoder first expresses the update in row-normalized code units and then alternates low-rank approximation with interval projection. Reconstruction requires only the transmitted codes, scales, and known batch size.

Row normalization. Let $\mathbf{D} = \operatorname{diag}(\mathbf{s})$. We express a candidate update $\mathbf{U}'$ in code units as

$$\mathbf{Z} = \mathbf{D}^{-1}\mathbf{U}'. \tag{4}$$

Because all scales are positive, the invertible row scaling preserves rank and row space. In value units, the entrywise quantization error in row i is bounded by $s_i/2$. Dividing each row by the corresponding scale gives the same error bound of $1/2$ for every entry. Row normalization therefore places the quantization constraints on a common scale before low-rank approximation.

Approximate alternating projections. After row normalization, the decoder combines a low-rank constraint with the quantization-interval constraints. Let $\mathcal{M}_r$ denote the set of matrices with rank at most r, where the assumed rank is $r = B$ unless stated otherwise, and let $\widetilde{\mathcal{C}} = \mathbf{D}^{-1}\mathcal{C}(\mathbf{Q}, \mathbf{s})$ denote the normalized interval box corresponding to Eq. (3). The iteration can be described as

$$\mathbf{Z}_k = \Pi_{\widetilde{\mathcal{C}}}\big(\Pi_{\mathcal{M}_r}(\mathbf{Z}_{k-1})\big). \tag{5}$$

The rank projection retains the dominant components of the current estimate. We use randomized SVD to efficiently compute a rank-r approximation. The subsequent interval

projection clips each entry to the bounds specified by the received code. Repeating low-rank approximation and interval projection allows the decoder to exploit both the update structure and the quantization constraints.

The implementation moves interval bounds inward by 10^{-6} to avoid rounding-boundary ambiguity. This shift also excludes the exact row-maximum values $\pm Q_b$, which changes the affected entries by at most 10^{-6} in code units. After K iterations, we can obtain rescaled row by the corresponding received scale as

$$\widehat{\mathbf{U}} = \mathrm{diag}(\mathbf{s})\mathbf{Z}_K, \tag{6}$$

and pass $\widehat{\mathbf{U}}$ to CPA. Algorithm 1 summarizes the procedure.

The final interval projection can increase rank, so the reconstructed update is not guaranteed to have rank at most r after a finite number of iterations. The proposed method does not guarantee a unique solution or exact recovery. We evaluate reconstruction quality and numerical constraint satisfaction experimentally.

Algorithm 1 Consistent low-rank (CLR) dequantization

Require: codes $\mathbf{Q}$, scales $\mathbf{s}$, assumed rank r (default $r = B$), iterations K

$\ell \leftarrow \max(\mathbf{Q} - \frac{1}{2}, -Q_b) + 10^{-6}$
$u \leftarrow \min(\mathbf{Q} + \frac{1}{2}, Q_b) - 10^{-6}$
$\mathbf{Z} \leftarrow \mathbf{Q}$
for $k = 1, \ldots, K$ **do**
  $\mathbf{Z} \leftarrow$ randomized rank-r approximation of $\mathbf{Z}$
  $\mathbf{Z} \leftarrow \min(\max(\mathbf{Z}, \ell), u)$ ▷ entrywise
end for
return $\mathrm{diag}(\mathbf{s})\mathbf{Z}$

V. Experiments

A. Experimental setup

We use a torchvision VGG16 checkpoint obtained after one FedAvg round across three clients, each performing five local epochs. The model is initialized with ImageNet-pretrained weights and a reinitialized classifier.

We use two image sets. The *disjoint set* consists of 160 randomly selected ImageNet validation images [12], divided into ten groups of 16 with no image appearing in more than one group (Sec. V-B). Each group represents one client's training batch. For each batch, we start from the same public checkpoint and compute a separate client update. The *overlapping set* consists of 17 batches formed from a separate set of 16 images and is used in Secs. V-C and V-D.

Similar to the VGG16 update-inversion setting of CPA [3], the client freezes the convolutional feature extractor and trains only the FC layers. In our experiments, the client performs three full-batch SGD steps on the same batch. The learning rate is 10^{-3}, with no momentum or weight decay, and dropout is disabled. Images are resized to 256, center-cropped to 224×224, and normalized using the ImageNet mean and standard deviation. Only the FC1 weight update, of size 4096×25088, is quantized.

TABLE I
Feature recovery on the disjoint set (ten batches of 16, 4-bit per-channel). Failure: matched cosine below 0.2.

Method	Mean cosine	Minimum	Failures
FP32	0.986	0.194	1/160
INT4 PC	0.827	0.064	27/160
PC + norm. SVD	0.884	0.049	18/160
PC + CLR	**0.965**	0.337	**0**/160

Compared methods. We compare four inputs to CPA. The FP32 baseline uses the unquantized update. Per-channel center decoding (INT4 PC) reconstructs the update by multiplying each integer code by the corresponding row scale. Row-normalized SVD (norm. SVD) first divides each row of the center-decoded update by the received scale, computes a rank-r approximation, and rescales the rows. This baseline applies low-rank approximation once without interval projection. CLR combines row normalization with repeated low-rank approximation and interval projection.

For each batch, all methods are evaluated against the same underlying client update. The three quantized methods use the same 4-bit per-channel codes and scales. We set the assumed rank to $r = B = 16$, use $K = 100$ iterations for CLR, and run CPA for 25,000 iterations for every method.

Evaluation metrics. We compare the absolute values of the recovered features with the true FC1 input features using cosine similarity, because ICA recovers sources only up to sign and scale. Because CPA does not preserve the order of the sources, Hungarian matching assigns each recovered feature to a different ground-truth feature by maximizing the total cosine similarity. We report the mean cosine across all source instances and count a feature-recovery failure when the matched cosine is below 0.2. Ground-truth features are used only for evaluation.

To quantify subspace distortion, we report the largest principal angle between the row space of $\mathbf{H}$ and the dominant B-dimensional right singular subspace of the update supplied to CPA. We also report the relative update error $\|\widehat{\mathbf{U}} - \mathbf{U}\|_F / \|\mathbf{U}\|_F$, where $\widehat{\mathbf{U}}$ is the decoded update.

B. Feature recovery across image batches

Table I compares feature recovery across the four methods. CLR increases the mean cosine from 0.827 under center decoding to 0.965 and reduces the number of failures from 27 to zero. The mean cosine improves in every batch. Row-normalized SVD also improves recovery over center decoding but leaves 18 failures. The additional improvement from CLR demonstrates the benefit of combining low-rank approximation with quantization-interval constraints.

The improvement does not imply uniformly near-perfect recovery. CLR's minimum cosine is 0.337, and three of its 160 sources remain below 0.65 (Fig. 1). At threshold 0.65, center decoding and normalized SVD have 28 and 18 failures, respectively, compared with three for CLR. Thus the

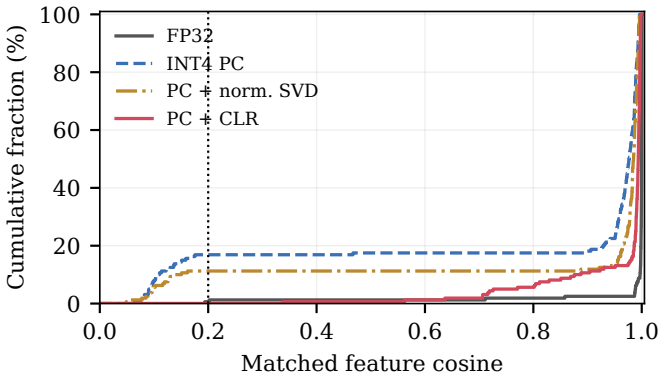


Fig. 1. Empirical cumulative distribution of matched feature cosine on the disjoint set (160 images). The dashed line marks the failure threshold 0.2. CLR removes failures by this criterion, but retains three sources below cosine 0.65.

TABLE II
FEATURE RECOVERY VERSUS BATCH SIZE ON THE OVERLAPPING SET (4 BIT). ENTRIES: MEAN COSINE / FAILURES. FP32 GIVES 0.999 / 0 FOR EVERY B.

B	Batches	INT4 PT	INT4 PC	PC + CLR
16	1	0.609 / 5 of 16	0.819 / 3 of 16	**0.986** / **0**
12	8	0.600 / 31 of 96	0.833 / 16 of 96	**0.987** / **0**
8	8	0.597 / 21 of 64	0.858 / 9 of 64	**0.989** / **0**

gain persists under this stricter threshold, although the zero-failure statement does not. Unlike the 4-bit results on the overlapping set (Secs. V-C and V-D), the disjoint set yields several intermediate-quality sources.

FP32 CPA averages 0.986 but has one failure, at cosine 0.194, in a batch whose largest principal angle is only 6.36°. Another batch has minimum cosine 0.202. Conversely, three CLR batches have angles above 60° without a failure at 0.2. These cases limit a purely subspace-based explanation: decoding, numerical conditioning, and source separation can all affect the outcome. They also show that removing quantization alone does not guarantee CPA success.

C. Recovery across batch sizes and image reconstruction

We examine the effect of batch size using the overlapping set. We evaluate its 16 images as one batch with $B = 16$. From the same image set, we also select eight distinct subsets with $B = 12$ and eight distinct subsets with $B = 8$. Each batch starts from the same public checkpoint, and the client update is computed separately for each image combination. The resulting 17 batches contain 176 image instances in total, but reuse the same 16 images.

We compare 4-bit per-tensor (PT) and per-channel (PC) quantization with standard center decoding. CLR is applied to the per-channel quantized updates, with FP32 updates serving as the unquantized baseline.

Table II shows that CLR improves feature recovery at every evaluated batch size. Across the 176 image instances, per-channel center decoding produces 28 failures, whereas CLR produces none. For the full 16-image batch, CLR reduces the relative update error from 0.340 to 0.093 and the largest principal angle from 89.7° to approximately 28°.

TABLE III
IMAGE RECONSTRUCTION ON THE 16-IMAGE BATCH (MEAN ± STD OVER THREE SEEDS). LAST COLUMN: IMAGE–SEED PAIRS WITH LPIPS ABOVE 0.7 (OF 48).

Method	rel. err.	PSNR↑	SSIM↑	LPIPS↓	> 0.7
FP32	0	12.27±.04	.343±.001	.509±.026	2
INT4 PT	.690	11.62±.02	.290±.004	.824±.002	48
INT4 PC	.340	11.73±.02	.289±.008	.697±.014	22
PC + CLR	.093	**12.11**±.03	**.333**±.001	**.557**±.004	**0**

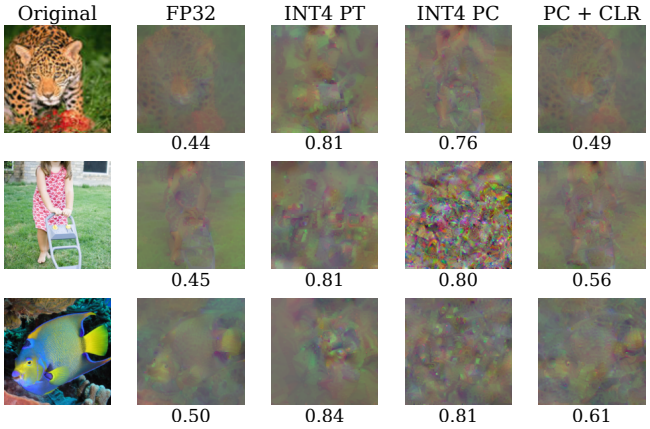


Fig. 2. Reconstructed images from the 16-image batch using initialization seed 42. The first column shows the original images, and numbers below the reconstructions indicate LPIPS. The examples are selected for large LPIPS improvements over per-channel center decoding. Rows are arranged by pixel-MSE matching and do not necessarily preserve source identity.

We evaluate image reconstruction on the full 16-image batch. For each recovered feature, an image is optimized so that the public feature extractor produces a feature vector matching the absolute recovered feature. We use Adam for 25,000 iterations with an initial learning rate of 0.1, a cosine learning-rate schedule, and a TV regularization weight of 0.1. The reconstruction procedure is repeated with three initialization seeds.

Because the reconstructed images have no fixed order, we use Hungarian matching to minimize the total pixel MSE between the reconstructed and ground-truth images. PSNR, SSIM, and VGG-LPIPS [13] are computed using the resulting image pairs. This matching evaluates the reconstructed image set and can differ from the feature-based matching used to evaluate source recovery.

Table III shows that the feature-recovery improvements also benefit image reconstruction under pixel-MSE matching. Relative to per-channel center decoding, CLR closes 70% of the PSNR gap, 82% of the SSIM gap, and 74% of the LPIPS gap to the FP32 baseline. The number of image–seed pairs with LPIPS above 0.7 decreases from 22 under per-channel center decoding to zero under CLR. The FP32 baseline produces two such pairs.

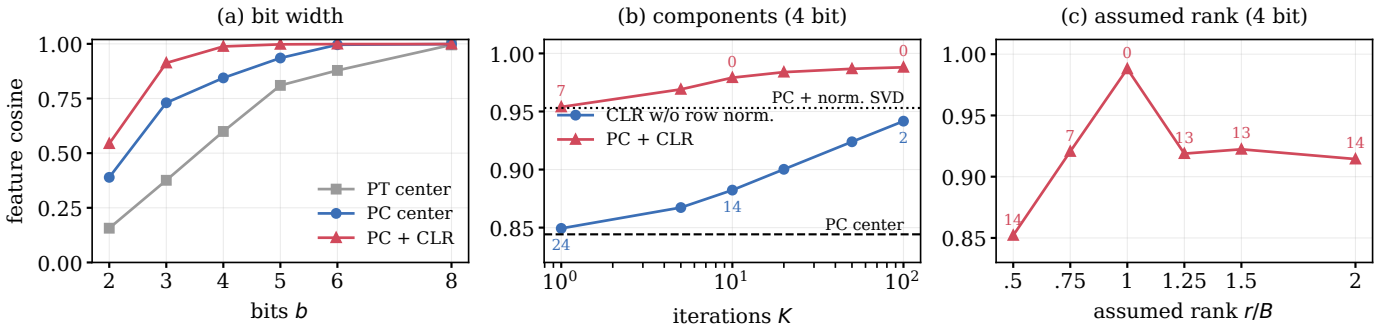


Fig. 3. Controlled sweeps on the overlapping set (17 batches, 176 image instances from 16 unique images). Cosines are unweighted means across batches; annotations count failures. (a) Bit width. (b) Components at four bits; CLR w/o row norm. alternates rank truncation and interval projection without row normalization, and horizontal lines mark single-pass baselines. (c) Assumed rank relative to B. These sweeps use the overlapping set only.

Fig. 2 illustrates examples with large LPIPS improvements over per-channel center decoding. The displayed reconstructions are arranged using pixel-MSE matching, so a candidate displayed beside a ground-truth image may correspond to a different source under feature-based matching. Although CLR improves image reconstruction metrics, the low PSNR values even under FP32 indicate that accurate feature recovery does not necessarily produce a faithful reconstruction of the original image.

D. Ablation, bit width, and assumed rank

Bit width. In Fig. 3a, at three bits, CLR's batch-mean cosine is 0.912 with no failures, exceeding four-bit center decoding (0.844, 28 failures) and approaching five-bit center decoding (0.935, 12 failures). At four bits, CLR reaches 0.988, between five- and six-bit center decoding (0.935 and 0.996). This is an empirical shift of roughly one to two bits along the feature-cosine curve of the overlapping set, not a measured communication saving, and it was not evaluated on the disjoint set. Two-bit CLR improves cosine from 0.389 to 0.544 but leaves 16 failures.

Components. Fig. 3b compares the effects of row normalization and interval projection. The x-axis shows the iteration count K, while the numbers beside the curves indicate feature-recovery failures among 176 image instances.

We can observe that, without row normalization, alternating low-rank approximation and interval projection reduces the failure count from 24 at $K = 1$ to two at $K = 100$. On the other hand, CLR combined with row normalization decreases the failure counts from seven at $K = 1$ to one at $K = 5$, and no failures after $K = 10$. These results demonstrate the additional benefit of enforcing quantization intervals after row normalization.

Assumed rank. In Fig. 3c, we can see that using $r \leq 0.75B$ discards useful information, while $r \geq 1.25B$ allows more quantization error in the fit; these choices leave 7–14 failures. Overestimating the rank by 25% already leaves 13 failures, more than the seven of row-normalized SVD at $r = B$, so reliable rank selection is necessary in practice.

VI. Conclusion

Quantization can degrade gradient inversion attacks, but the degradation observed under standard dequantization does not fully reflect the information available to an attacker. We proposed consistent low-rank dequantization to exploit the low-rank structure of FC updates together with the intervals specified by the received codes and scales. Experiments demonstrate that CLR improves CPA feature recovery using only the transmitted codes, scales, and known batch size. The results show that information obscured by standard dequantization can remain useful when an attacker exploits the update structure and quantization constraints. Privacy protection should therefore be assessed against structure-aware decoding rather than inferred solely from attacks applied to standard dequantized updates. The current evaluation uses a frozen feature extractor, repeated training on the same batch, and a known batch size as the assumed rank. Future work will examine changing batches, trainable feature extractors, and rank selection from the packet alone, where the batch-size rank bound may no longer hold.

References

[1] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 2019.

[2] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" *Advances in neural information processing systems*, vol. 33, pp. 16 937–16 947, 2020.

[3] S. Kariyappa, C. Guo, K. Maeng, W. Xiong, G. E. Suh, M. K. Qureshi, and H.-H. S. Lee, "Cocktail party attack: Breaking aggregation-based privacy in federated learning using independent component analysis," in *International Conference on Machine Learning*. PMLR, 2023, pp. 15 884–15 899.

[4] P. R. Ovi, E. Dey, N. Roy, and A. Gangopadhyay, "Mixed quantization enabled federated learning to tackle gradient inversion attacks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023, pp. 5046–5054.

[5] K. Yue, R. Jin, C.-W. Wong, D. Baron, and H. Dai, "Gradient obfuscation gives a false sense of security in federated learning," in *32nd USENIX security symposium (USENIX Security 23)*, 2023, pp. 6381–6398.

[6] R. Zhang, S. Guo, J. Wang, X. Xie, and D. Tao, "A survey on gradient inversion: Attacks, defenses and future directions," *arXiv preprint arXiv:2206.07284*, 2022.

[7] Z. Li, J. Zhang, L. Liu, and J. Liu, "Auditing privacy defenses in federated learning via generative gradient leakage," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 10 122–10 132.

[8] X. Ding, Z. Liu, X. You, X. Li, and A. V. Vasilakos, "Improved gradient leakage attack against compressed gradients in federated learning," *Neurocomputing*, vol. 608, p. 128349, 2024.

[9] N. T. Thao and M. Vetterli, "Deterministic analysis of oversampled A/D conversion and decoding improvement based on consistent estimates," *IEEE Transactions on Signal Processing*, vol. 42, no. 3, pp. 519–531, 1994.

[10] V. K. Goyal, M. Vetterli, and N. T. Thao, "Quantized overcomplete expansions in $\mathbb{R}^N$: Analysis, synthesis, and algorithms," *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 16–31, 1998.

[11] A. Moshtaghpour, L. Jacques, V. Cambareri, K. Degraux, and C. De Vleeschouwer, "Consistent basis pursuit for signal and matrix estimates in quantized compressed sensing," *IEEE Signal Processing Letters*, vol. 23, no. 1, pp. 25–29, 2016.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[13] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *2018 IEEE/CVF conference on computer vision and pattern recognition*. IEEE, 2018, pp. 586–595.