Deep Leakage from Gradients

Ligeng Zhu, Zhijian Liu and Song Han

摘要

在现代多节点机器学习系统中（例如，分布式训练和协作学习），梯度交换是一种广泛使用的方法。长期以来，人们一直认为梯度共享是安全的，即通过梯度交换不会泄露训练数据。然而，本文展示了可以从公开共享的梯度中获取私有训练数据的可能性。本文将此种泄露称为梯度深度泄露，并在计算机视觉和自然语言处理任务上通过实证验证了其有效性。实验结果表明，所提出的攻击方法比以前的方法更为强大：对于图像，恢复是像素级精确的，对于文本则是词汇级匹配的。因此，本文希望提高人们对梯度安全性的认识。本文还讨论了几种可能的策略来防止这种深度泄露。在不改变训练设置的情况下，最有效的防御方法是梯度剪枝。

关键词：联邦学习；隐私泄露；梯度重构；深度学习

1 引言

分布式训练已成为加快大规模数据集训练的关键技术。在这一系统中，各工作节点并行执行计算任务，并与参数服务器 [1] [2] 进行梯度交换或全归约同步 [3]。这种计算方式自然形成数据分割：每个节点拥有独立的训练数据，训练期间仅通过梯度交流，不涉及数据中心 [4]。这允许不同来源的数据合作训练模型，尤其在涉及隐私信息时广泛应用，如多家医院在不共享病患数据的情况下共同训练模型 [5]。

尽管分布式训练和协作学习在机器学习中应用广泛，但梯度共享是否能保护数据隐私仍是疑问。以往认为梯度较为安全，难以泄露训练数据。但最新研究显示，梯度能够泄露训练数据的特定属性 [6]。本文探讨是否可能从梯度中完整获取训练数据。对于已知的机器学习模型 $F()$ 和权重 W，探究是否能通过梯度 ∇w 反推出训练数据。

本研究证明了通过共享梯度能泄露私密训练数据，同时提出一种算法，仅经过少量迭代即可复原输入和标签。通过先生成一对随机的“假”输入和标签，进行正常的前向和反向过程，得到虚拟梯度后，优化这对虚拟数据，以缩小其与真实梯度的差距。优化完成时，私密训练数据便被曝光。

与传统的“浅层”泄露相比，该方法不需额外信息即可揭示训练数据，而不是类似的合成图像 [7]。本文首次发现了从梯度中的深层泄露，并希望提高对梯度安全性重新评估的意识。

深层泄露为多节点机器学习系统的隐私保护带来挑战。本文的发现表明，梯度共享并非总能可靠地保护数据隐私。在集中式分布式训练中，参数服务器无需存储数据即可窃取信息。在去中心化训练中，情况更为严峻，任何节点均可能盗取数据。本文提出了三种防御策略：梯

度扰动、低精度表示和梯度压缩。发现一定程度的噪声可有效防御，而梯度压缩通过剪枝可成功抵御攻击。

本文的贡献包括：

- 证明从公开梯度中获取私有数据的可能性，DLG 算法在此领域为首例。
- 不同于依赖额外信息的传统方法，DLG 依靠梯度揭示高精度图像和匹配文本。
- 分析了不同情境下攻击的难度，讨论了多种防御策略。

2 相关工作

2.1 分布式训练

训练大规模机器学习模型，如深度神经网络，需要大量计算资源。为缩短训练时间，众多研究致力于提升分布式训练效率。这些努力涵盖了算法优化 [8] 和框架开发 [9]，以增强其扩展能力。同步 SGD 由于扩展时的性能稳定性，成为多数研究的基础。分布式训练主要分为两种类型：具有参数服务器的集中式训练 [1] [2] [10]，以及无参数服务器的去中心化训练 [3] [9]。在这两种架构下，节点首先独立计算并更新本地权重，随后交换梯度。集中式模式下，梯度先汇总后再分配至各节点；去中心化模式中，梯度直接在邻节点间传递。鉴于许多训练数据的隐私性，协作学习 [5] [11] 应运而生，允许多方参与者在不共享原始数据集仅共享梯度的情况下共同训练模型。这种方法已被应用于多医院合作训练医疗模型 [11]，跨国分析病人生存率 [5]，以及开发预测性键盘以优化打字体验 [12]。

2.2 梯度的浅泄露

先前的研究已经在从梯度中提取训练数据信息方面取得了进展。在某些层，梯度暴露了一定量的数据。例如，在语言处理中，只有训练集中出现的词汇才会在嵌入层上产生梯度，这间接泄露了训练数据中的词汇使用情况 [6]。然而，这类信息泄露是有限的：泄露的词汇是无序的，且由于语义歧义，重建原始句子颇具挑战。在全连接层，通过分析梯度更新可以推测出输出特征，但这种方法并不适用于卷积层，因为特征的维数通常远超过权重的规模。最新的研究尝试采用基于学习的方法来推断批处理数据的特性。研究者们证明了，通过在梯度上训练的二元分类器，可以判定特定的数据记录是否存在于参与方的批次中（成员身份推断 [13]）或是否含有特定属性（属性推断 [6]）。此外，研究人员利用 GAN 模型 [7] 从梯度中生成与训练数据类似的图像 [14]。然而这种攻击手段有其局限性，只在数据记录在视觉上具有高度一致性时（例如在面部识别任务中）才有效。

3 本文方法

本文展示了如何像素级别地窃取图像和逐词窃取句子中的梯度，如算法 1所示。在每一步 t，每个节点 i 从其数据集中采样一个小批量 $(\mathbf{x}_{t,i}, \mathbf{y}_{t,i})$ 来计算梯度

$$\nabla W_{t,i} = \frac{\partial \ell(F(\mathbf{x}_{t,i}, W_t), \mathbf{y}_{t,i})}{\partial W_t} \tag{1}$$

Algorithm 1 Deep Leakage from Gradients.

Input: $F(\mathbf{x}; W)$: Differentiable machine learning model; W: parameter weights; ∇W: gradients calculated by training data

Output: Private training data $\mathbf{x}, \mathbf{y}$

1: **procedure** DLG$(F, W, \nabla W)$
2: $\quad \mathbf{x}' \leftarrow \mathcal{N}(0,1)$, $\mathbf{y}' \leftarrow \mathcal{N}(0,1)$ ▷ Initialize dummy inputs and labels.
3: $\quad$ **for** $i = 1$ to n **do**
4: $\quad\quad \nabla W_i' \leftarrow \partial \ell(F(\mathbf{x}'_i, W_t), \mathbf{y}'_i)/\partial W_t$ ▷ Compute dummy gradients.
5: $\quad\quad \mathbb{D}_i \leftarrow \|\nabla W_i' - \nabla W\|^2$
6: $\quad\quad \mathbf{x}'_{i+1} \leftarrow \mathbf{x}'_i - \eta \nabla_{\mathbf{x}'_i} \mathbb{D}_i, \mathbf{y}'_{i+1} \leftarrow \mathbf{y}'_i - \eta \nabla_{\mathbf{y}'_i} \mathbb{D}_i$ ▷ Update data to match gradients.
7: $\quad$ **end for**
8: $\quad$ **return** $\mathbf{x}'_{n+1}, \mathbf{y}'_{n+1}$
9: **end procedure**

梯度在 N 个服务器之间平均，然后用于更新权重：

$$\overline{\nabla W_t} = \frac{1}{N} \sum_{j}^{N} \nabla W_{t,j}; \quad W_{t+1} = W_t - \eta \overline{\nabla W_t} \tag{2}$$

给定从其他节点 k 处接收的梯度 $\nabla W_{t,k}$，我们旨在窃取参与者 k 的训练数据 $(\mathbf{x}_{t,i}, \mathbf{y}_{t,i})$。请注意，$F()$ 和 W_t 默认是同步分布式优化中共享的。

为了从梯度中恢复数据，我们首先随机初始化一个虚拟输入 $\mathbf{x}_{t,i}$ 和一个虚拟标签 $\mathbf{y}_{t,i}$（如算法 1的第 2 行所示）。然后我们将这些“虚拟数据”输入模型中并得到“虚拟梯度”：

$$\nabla W' = \frac{\partial \ell(F(\mathbf{x}', W), \mathbf{y}')}{\partial W} \tag{3}$$

通过优化虚拟梯度使其尽可能接近原始梯度，也使虚拟数据接近实际训练数据（如图 1所示的趋势）。给定在某一步得到的梯度，我们通过最小化下面的目标函数来得到训练数据

$$\mathbf{x}'^*, \mathbf{y}'^* = \arg\min_{\mathbf{x}', \mathbf{y}'} \|\nabla W' - \nabla W\|^2 = \arg\min_{\mathbf{x}', \mathbf{y}'} \|\frac{\partial \ell(F(\mathbf{x}', W), \mathbf{y}')}{\partial W} - \nabla W\|^2 \tag{4}$$

距离 $||\nabla W' - \nabla W||^2$ 关于虚拟输入 $\mathbf{x}'$ 和标签 $\mathbf{y}'$ 是可微分的，因此可以使用标准的基于梯度的方法来优化。请注意，这种优化需要二阶导数。我们做了一个温和的假设，即 F 是二次可微分的，这对于大多数现代机器学习模型（例如，大多数神经网络）和任务是成立的。

4 复现细节

4.1 与已有开源代码对比

本文介绍的代码已开源。本次复现工作不仅再现了原始代码，还旨在通过拓展改进及延伸，特别是将其应用于交替方向乘子法（ADMM）[15]。ADMM 是一种适用于带约束的二分块凸优化问题的分布式算法，它通过将原始问题分解为多个子问题来求解。与梯度下降法（GD）相比，ADMM 不仅具有更温和的收敛条件，而且能有效处理异质性问题，尤其在联邦学习研

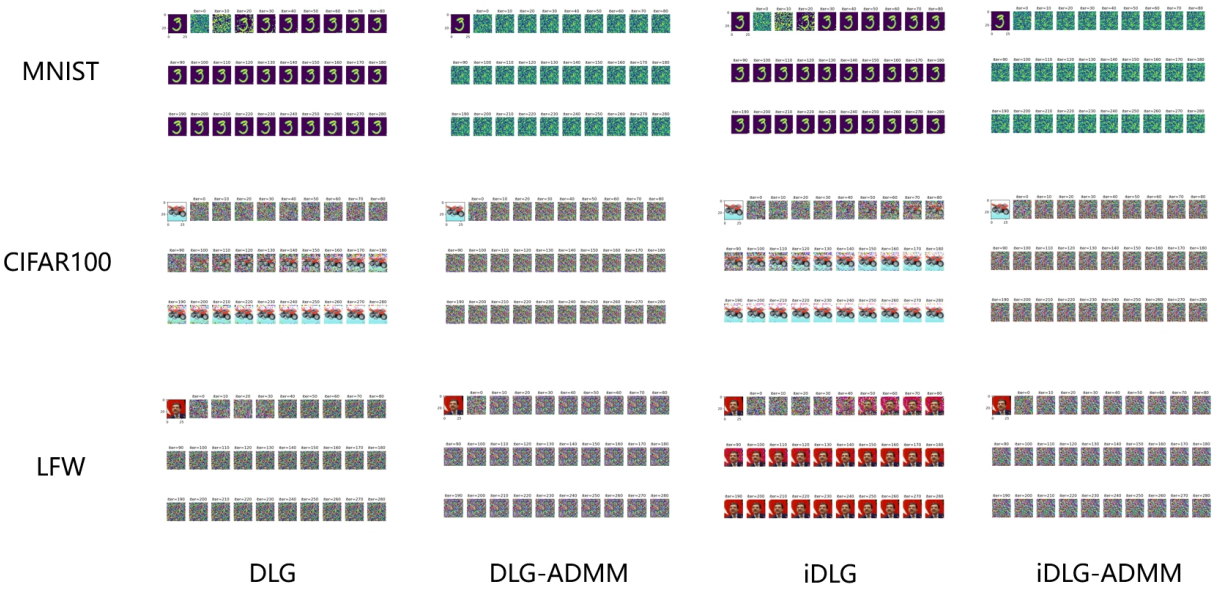

图 1. 三个数据集下四种算法重构图片的结果

究中显示出其高效的通信性能。由于 ADMM 在达到最优解时只需较少的通信次数，直观上，它传递的参数可能包含比 SGD 更丰富的信息，这引发了一个问题：ADMM 是否比 SGD 更有可能泄露隐私？

目前，关于隐私保护的大多数研究强调 ADMM 可能导致隐私泄露，从而对 ADMM 进行改进 [16] [17] [18]。然而，目前尚无文献明确表明 ADMM 会泄露隐私。不同于一阶下降求解器如梯度下降，ADMM 作为一种道格拉斯分裂法，它在更新每个子问题的变量时，并不依赖于目标函数的梯度。基于这一事实，笔者对 ADMM 泄露隐私这一广泛接受的观点提出了质疑，并计划通过实验来探索 ADMM 泄露隐私的程度，以及它是否确实比 SGD 算法暴露更多的隐私信息。

4.2 创新点

4.2.1 算法改进

由于 DLG 更新基于梯度下降法，而 ADMM 基于原始对偶方法，主要传递参数而非梯度，使得 ADMM 难以直接应用于 DLG。为与 SGD 的 DLG 相比较，本研究首先借鉴了传递增量的 ADMM 算法，即 FedADMM [19]。随后，通过将增量视作梯度的近似，将二者结合，提出了一种基于 ADMM 的 DLG 算法——DLG-ADMM，详见算法 2。

4.2.2 代码改进

由于 ADMM 代码的开源较为有限，相比于 SGD 中直接使用 torch.grad，笔者重新编写了 ADMM 的更新部分，并对冗余代码进行了优化和简化。同时，为了进一步与 SGD 算法进行对比，笔者将 iDLG 算法 [20] 与 ADMM 结合，构建了 iDLG-ADMM 算法。为了实现与原论文中 LFW 数据集算法的比较，笔者还重新编写了数据导入类，使得算法能够使用 CIFAR100 和 MNIST10 以外的数据集。

Algorithm 2 DLG-ADMM

Input: Maximum number of iterations K, local epoch number E_i, client learning rate η_i, batches B, ∇_i^k: gradients calculated by training data
Output: private training data x, y
Malicious i **Do:**
$x_i' \leftarrow \mathcal{N}(0,1)$; $y_i' \leftarrow \mathcal{N}(0,1)$
for $k = 0$ **to** $K-1$ **do**
    $\nabla_i^{k'} \leftarrow \text{ClientUpdate}(x_i', y_i')$
    $D_i \leftarrow \|\nabla_i^{k'} - \nabla_i^k\|^2$
    $x_i^{k+1} \leftarrow x_i^k - \eta_i \nabla_x D_i$
    $y_i^{k+1} \leftarrow y_i^k - \eta_i \nabla_y D_i$
end for
procedure ClientUpdate(x, y)
    for $t = 0$ **to** $E_i - 1$ **do**
        for each batch $b \in B$ **do**
            Compute $\nabla f_i(x_i, b)$
            $x_i^{t+1} \leftarrow x_i^t - \eta_i(\nabla f_i(x_i, b) + \pi_i + \rho(x_i^t - z_i^t))$
        end for
    end for
    $\pi_i^{k+1} \leftarrow \pi_i^k + \rho(x_i^{k+1} - z_i^k)$
    $\Delta_i^k \leftarrow (x_i^{k+1} + \rho^{-1}\pi_i^{k+1}) - (x_i^k + \rho^{-1}\pi_i^k)$
    $z_i^{k+1} \leftarrow z_i^k + \eta_i \Delta_i^k$
    $\nabla_i^k \leftarrow -\Delta_i^k / \eta_i$
end procedure

5 实验结果分析

图 1展示了使用 DLG 和 iDLG 技术从梯度中重构出的原始图片，以及使用基于 ADMM 方法的 DLG-ADMM 和 iDLG-ADMM 技术的结果。由图可见，DLG 和 iDLG 技术在 CIFAR100、MNIST 和 LFW 数据集上重构的图像随迭代次数的增加变得越来越清晰。这表明当使用 DLG 或 iDLG 方法时，隐私信息（即原始训练图片）可能会从梯度中泄露出来。相比之下，基于 ADMM 方法的 DLG-ADMM 和 iDLG-ADMM 技术的重构图像，在所有迭代步骤中都维持着较高的噪声水平，使得原始图片难以辨认。这表明 ADMM 方法可能在某种程度上保护了训练数据的隐私，因为即使在多次迭代后，也难以从梯度中恢复出原始数据。

在 DLG 和 iDLG 技术下，我们可以看到随着迭代次数的增加（从 iter=0 到 iter=280），重构出的图片逐渐从一开始的噪声图像变为清晰可辨的图像。这种变化表明，随着优化过程的进行，梯度中包含的关于原始图片的信息被逐步“解码”。而基于 ADMM 的方法并不会随着迭代的增加而被重构图片。这一定程度的说明了 ADMM 算法可能是一种有效的隐私保护技术，特别是在需要防止敏感数据泄露的联邦学习环境中。

笔者还通过探究算法性能与迭代次数之间的关系来评估算法效果和收敛性，特别是在

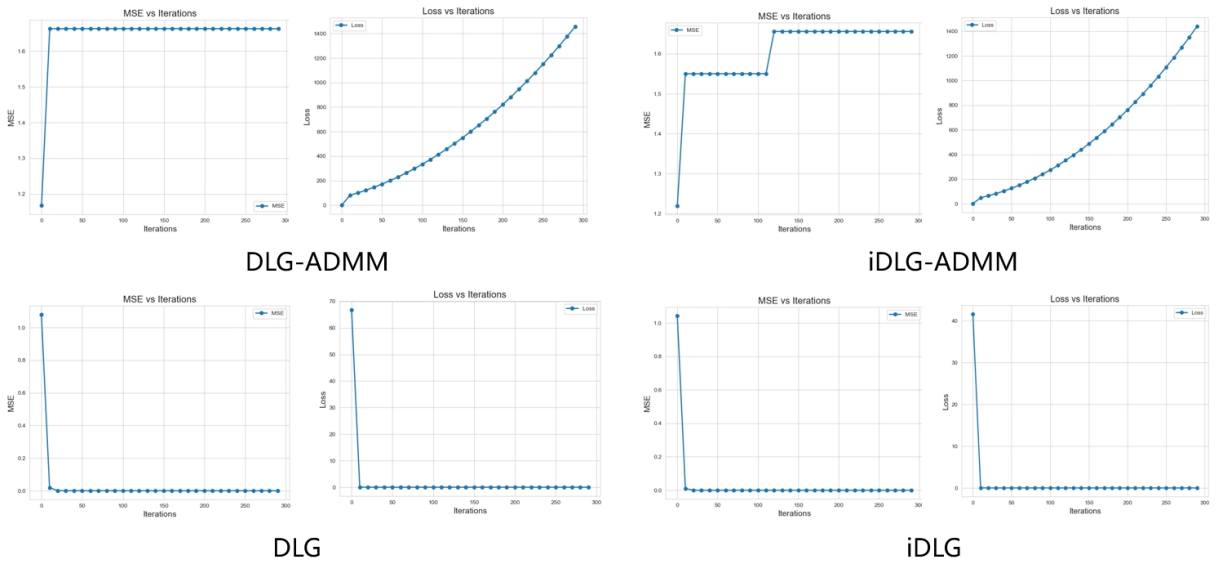

图 2. 四种算法在 MNIST 上的均方误差与损失值随迭代次数增加的变化趋势

MNIST 数据集上的均方误差（MSE）和损失表现，如图 2所示。实验结果显示，其他两个数据集的性能和损失趋势与 MNIST 数据集相似，因此，为节约篇幅，仅展示了 MNIST 数据集的结果。图中可以看出，DLG 算法的 MSE 迅速降至近零水平并保持稳定，表明重构质量较高；损失在初期下降后稳定在低水平，说明优化过程迅速收敛。相对而言，ADMM 算法的损失随迭代次数增加而线性上升，表明其优化性能较弱，但可能提供更强的隐私保护。这一现象也间接反映了基于梯度下降的 DLG 攻击方法对 ADMM 算法的局限性。

6 总结与展望

本研究以 DLG 算法为起点，考察了基于梯度下降的攻击算法是否适用于 ADMM 算法。实验显示，与基于 ADMM 的方法相比，基于梯度的方法可能更容易遭受隐私泄漏，导致图像重构。鉴于 DLG 的攻击策略对 ADMM 无效，未来研究将专门为 ADMM 设计攻击算法。同时，实验发现，不同图像的恢复结果各异，DLG 类攻击的成功率受随机种子影响极大，显示出算法的鲁棒性不足。据此，未来研究可深入探讨这类算法的内在机制。

参考文献

[1] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, and Kurt Keutzer. Firecaffe: Near-linear acceleration of deep neural network training on compute clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2592–2600, 2016.

[2] Mu Li, David G. Andersen, Jun Woo Park, Alexander J. Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J. Shekita, and Bor-Yiing Su. Scaling distributed machine learning with the parameter server. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pages 583–598, 2014.

[3] Pitch Patarasuk and Xin Yuan. Bandwidth optimal all-reduce algorithms for clusters of workstations. *Journal of Parallel and Distributed Computing*, 69(2):117–124, 2009.

[4] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[5] Arthur Jochems, Timo M Deist, Issam El Naqa, Marc Kessler, Chuck Mayo, Jackson Reeves, Shruti Jolly, Martha Matuszak, Randall Ten Haken, Johan van Soest, et al. Developing and validating a survival prediction model for nsclc patients through distributed learning across 3 countries. *International Journal of Radiation Oncology* Biology* Physics*, 99(2):344–352, 2017.

[6] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706. IEEE, 2019.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[8] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, and Ke Yang. Large scale distributed deep networks. *Advances in neural information processing systems*, 25, 2012.

[9] Alexander Sergeev and Mike Del Balso. Horovod: fast and easy distributed deep learning in tensorflow. *arXiv preprint arXiv:1802.05799*, 2018.

[10] Arthur Jochems, Timo M. Deist, Johan Van Soest, Michael Eble, Paul Bulens, Philippe Coucke, Wim Dries, Philippe Lambin, and Andre Dekker. Distributed learning: Developing a predictive model based on data from multiple hospitals without data leaving the hospital– a real life proof of concept. *Radiotherapy and Oncology*, 121(3):459–467, 2016.

[11] Arthur Jochems, Timo M Deist, Johan Van Soest, Michael Eble, Paul Bulens, Philippe Coucke, Wim Dries, Philippe Lambin, and Andre Dekker. Distributed learning: developing a predictive model based on data from multiple hospitals without data leaving the hospital– a real life proof of concept. *Radiotherapy and Oncology*, 121(3):459–467, 2016.

[12] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[13] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[14] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

[15] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[16] Zonghao Huang, Rui Hu, Yuanxiong Guo, Eric Chan-Tin, and Yanmin Gong. Dp-admm: Admm-based distributed learning with differential privacy. *IEEE Transactions on Information Forensics and Security*, 15:1002–1012, 2019.

[17] Fanhua Shang, Tao Xu, Yuanyuan Liu, Hongying Liu, Longjie Shen, and Maoguo Gong. Differentially private admm algorithms for machine learning. *IEEE Transactions on Information Forensics and Security*, 16:4733–4745, 2021.

[18] Jiahao Ding, Xinyue Zhang, Mingsong Chen, Kaiping Xue, Chi Zhang, and Miao Pan. Differentially private robust admm for distributed machine learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1302–1311. IEEE, 2019.

[19] Yonghai Gong, Yichuan Li, and Nikolaos M Freris. Fedadmm: A robust federated deep learning framework with adaptivity to system heterogeneity. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 2575–2587. IEEE, 2022.

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