

# A Stealthy Wrongdoer: Feature-Oriented Reconstruction Attack against Split Learning *Supplementary Material*

Xiaoyang Xu<sup>1</sup> Mengda Yang<sup>1</sup> Wenzhe Yi<sup>1</sup> Ziang Li<sup>1</sup> Juan Wang<sup>1\*</sup> Hongxin Hu<sup>2</sup>  
Yong Zhuang<sup>1</sup> Yaxin Liu<sup>1</sup>

<sup>1</sup> Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education,  
School of Cyber Science and Engineering, Wuhan University

<sup>2</sup> Department of Computer Science and Engineering, University at Buffalo, SUNY

{xiaoyangx, mengday, wenzhey, ziangli, yong.zhuang, yaxin.liu}@whu.edu.cn

jwang@whu.edu.cn, hongxinh@buffalo.edu

## A. Experimental Setup Details

In this section, we provide more details about our experimental setup.

### A.1. Datasets

We elaborate on the four datasets used in our main experiments and Tab. 1 shows the details of datasets partitioning.

**CIFAR-10 [9].** CIFAR-10 is a classification benchmark dataset comprising 60,000  $3 \times 32 \times 32$  images categorized into 10 classes. It features 50,000 training images and 10,000 testing images, evenly distributed among the classes.

**CINIC-10 [2].** CINIC-10 extends CIFAR-10 by adding downsampled ImageNet samples in the same classes with CIFAR-10. Both datasets share the same classes, but CINIC-10 consists of 270,000  $3 \times 32 \times 32$  images. In comparison to CIFAR-10, CINIC-10 presents a more complex and diverse distribution.

**CelebA [12].** CelebA is a dataset related to facial attribute classification. It includes 202,599 facial images from 10,177 different celebrities, and each image is associated with 40 different attribute labels. In our experiment, we resize the images in the CelebA to  $3 \times 64 \times 64$ .

**FFHQ [8].** FFHQ was originally designed as a benchmark for Generative Adversarial Networks which contains 70,000 facial images. The dataset exhibits rich diversity and noticeable variations in terms of age, ethnicity, and image backgrounds. As well as CelebA, we resize the images in FFHQ to  $3 \times 64 \times 64$ .

Table 1. Details of the partitioning among different datasets.

Target Model	Target Dataset	Train	Test	Auxiliary Dataset	Image Size
MobileNet	CIFAR-10	50000	10000	CINIC-10 (5000)	$3 \times 32 \times 32$
ResNet-18	CelebA	162770	19962	FFHQ (10000)	$3 \times 64 \times 64$

### A.2. Model Architectures

The detailed model architectures of the substitute client model, inverse network, and discriminator are shown in Table 2. And Fig. 1 illustrates the different split strategies towards the target model.

## B. Additional Data Reconstruction Results

We present more reconstruction results of FORA in this section.

---

\*Corresponding author.

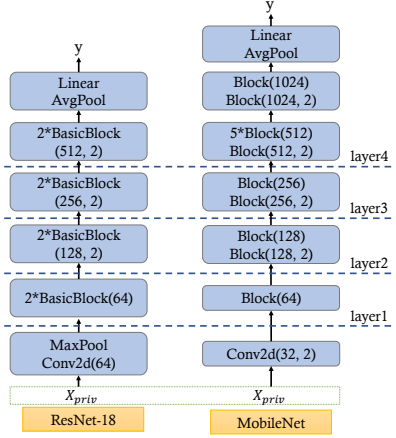


Figure 1. Target models splitting settings in our experiments.

Table 2. Architectures of the substitute client, inverse network, and discriminator for two datasets in different splitting points.

Split Point	CIFAR-10			CelebA		
	$\hat{F}_c$	$f_c^{-1}$	$D$	$\hat{F}_c$	$f_c^{-1}$	$D$
layer 1	2*Conv2d(32) MaxPool	ConvTrans(256, 2) Conv2d(256) Tanh	Conv2d(128, 2) Conv2d(256, 2) 6*ResBlock(256) Conv2d(256, 2) Linear	2*Conv2d(64) MaxPool	ConvTrans(256, 2) Conv2d(256) Tanh	Conv2d(128, 2) 2*Conv2d(256, 2) 6*ResBlock(256) Conv2d(256, 2) Linear
layer 2	2*Conv2d(64) MaxPool	ConvTrans(256, 2) Conv2d(256) Tanh	Conv2d(128, 2) Conv2d(256, 2) 6*ResBlock(256) Conv2d(256, 2) Linear	2*Conv2d(64) MaxPool	ConvTrans(256, 2) Conv2d(256) Tanh	Conv2d(128, 2) 2*Conv2d(256, 2) 6*ResBlock(256) Conv2d(256, 2) Linear
layer 3	2*Conv2d(64) MaxPool 2*Conv2d(128) MaxPool	ConvTrans(256, 2) ConvTrans(128, 2) Conv2d(128) Tanh	Conv2d(256, 2) 6*ResBlock(256) Conv2d(256, 2) Linear	2*Conv2d(64) MaxPool 2*Conv2d(128) MaxPool	ConvTrans(256, 2) ConvTrans(128, 2) Conv2d(128) Tanh	2*Conv2d(256, 2) 6*ResBlock(256) Conv2d(256, 2) Linear
layer 4	2*Conv2d(64) MaxPool 2*Conv2d(128) MaxPool 3*Conv2d(256) MaxPool	ConvTrans(256, 2) ConvTrans(128, 2) ConvTrans(64, 2) Conv2d(64) Tanh	Conv2d(256) 6*ResBlock(256) Conv2d(256, 2) Linear	2*Conv2d(64) MaxPool 2*Conv2d(128) MaxPool 3*Conv2d(256) MaxPool	ConvTrans(256, 2) ConvTrans(128, 2) ConvTrans(64, 2) Conv2d(64) Tanh	Conv2d(256, 2) Conv2d(256) 6*ResBlock(256) Conv2d(256, 2) Linear

### B.1. Comparison with Semi-Honest Attacks

The full results of the comparison between UnSplit, PCAT, and FORA are presented in Fig. 2 and Fig. 3.

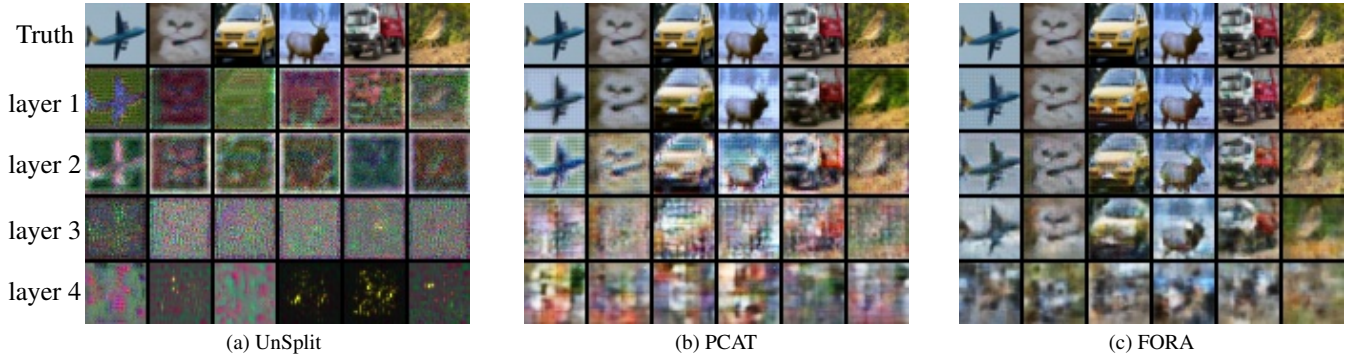


Figure 2. Additional results of Unsplit, PCAT and FORA on CIFAR-10 at all split points.

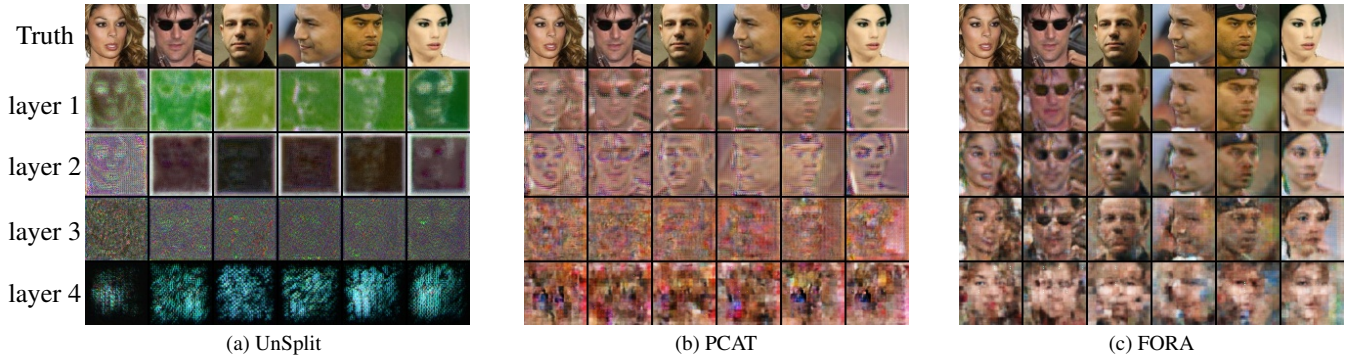


Figure 3. Additional results of Unsplit, PCAT and FORA on CelebA at all split points.

## B.2. Effect of Auxiliary Dataset

Then we present more detailed results of the impact of the auxiliary dataset on FORA. Fig. 4 presents complete results for the absence of categories on FORA. Fig. 5 displays additional results regarding the impact of the size of auxiliary datasets on FORA. Fig. 6 shows the overall results for the impact of auxiliary datasets distributions on FORA.

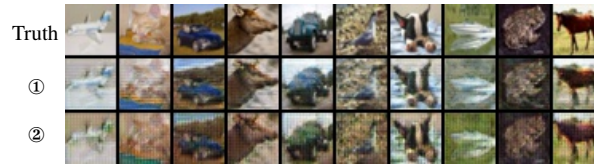


Figure 4. Additional results of the absence of categories on CIFAR-10. Row ① means absence class of living. Row ② means absence class of non-living.

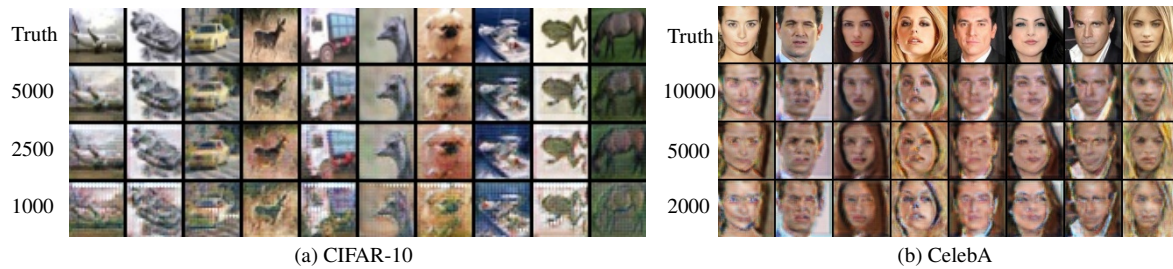


Figure 5. Additional results of varying auxiliary data size on FORA performed on CIFAR-10 and CelebA.



Figure 6. Additional results of auxiliary dataset distribution shift on FORA performed on CIFAR-10 and CelebA.

## B.3. Effect of Substitute Client Structure

In addition to the quantified results in the main text, we also present the reconstructed images by FORA under different substitute client model architectures in Fig. 7.

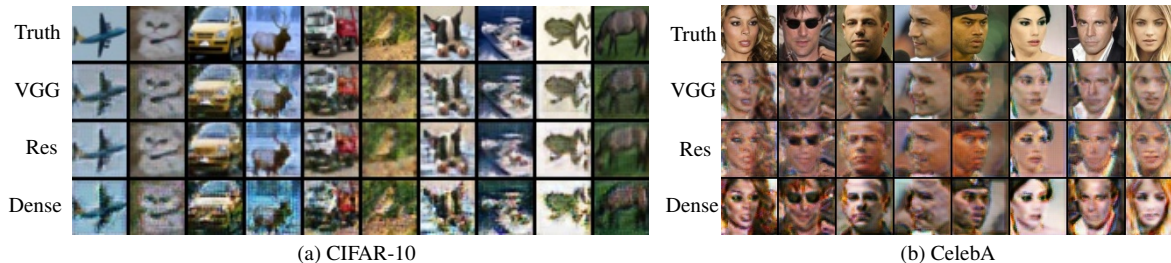


Figure 7. Additional results for FORA with varying substitute model architectures on CIFAR-10 and CelebA.

## C. Defense Techniques

In this section, we first provide a detailed introduction to the several defense mechanisms evaluated in our experiments. Then we present the additional defense results of CelebA and CIFAT-10. We finally discuss two possible adaptive defenses against our proposed method FORA.

### C.1. Defense Details

**Gradients Scrutinizer.** Gradients Scrutinizer (GS) [6] is a defense method against the malicious attacker FSHA [13]. In normal SL, gradients returned by servers exhibit greater similarity within the same label compared to gradients from different labels. However, in FSHA the client is trained as the encoder of an autoencoder to reconstruct training data without using target labels. As a result, gradients received by the client will not show notable distinctions between the same and different classes in FSHA. Based on this intrinsic difference, GS first computes the cosine similarity of gradients with the same label and those of different labels in each batch according to the received gradients. Subsequently, GS will calculate decision scores from three aspects: set gap, fitting error, and overlapping ratio, to distinguish hijacking servers from honest servers. If the detection score is above a set threshold, it is considered a normal SL. If it falls below the threshold, it is identified as a hijacking attack, and the training is stopped immediately.

The detection mechanism of GS depends on the model’s classification ability. Therefore, in the early stages of GS, some batches should be skipped to avoid the detection being misguided by the model that has not been well-fitted. Following this mechanism, we start GS at the 450th iteration in our experimental setup.

**Distance Correlation Minimization.** Distance Correlation [15, 18, 19] is a defense method widely used in SL to measure and mitigate the correlation between smashed data and the raw input, thereby preventing server adversaries from reconstructing the original input data. The loss function for this approach is as follows:

$$\mathcal{L} = \alpha \cdot DCOR(x, F_c(x)) + (1 - \alpha) \cdot TASK(y, F_s(F_c(x))) \quad (1)$$

where  $DCOR$  represents the distance correlation metric, and  $TASK$  denotes the classification loss between the true label  $y$  and the model’s prediction. By jointly minimizing the above loss, a better tradeoff can be achieved between preserving input data privacy and maintaining the utility of the model.

**Differential Privacy.** Differential Privacy was initially introduced to provide privacy guarantees for algorithms on aggregate databases [4, 5], and it was later applied to deep learning through DP-SGD [1]. Differential privacy has found widespread usage in various scenarios [10, 14], not exclusively in SL. Following the approach described in PCAT [7], we employ DP-SGD in the client model. Specifically, the client receives gradients from the server, clips each gradient using a threshold value  $C$ , and adds random noise to it. The client then utilizes these protected gradients to update its model, thereby safeguarding the privacy of the subsequent smashed data transmitted to the server. Different combinations of the clipping threshold  $C$  and noise scale  $\sigma$  yield varying privacy budgets  $\epsilon$  and levels of accuracy reduction.

**Noise Obfuscation.** Titcombe et al. [16] proposed an approach where additive Laplacian noise was directly added to smashed data before transmitting it to the server to defend against input reconstruction. This randomness introduces a higher level of complexity for adversaries, making it more challenging for them to infer the mapping between the smashed data and the private input.


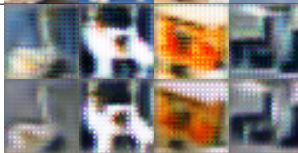
### C.2. More Defense Results and Possible Adaptive Defenses

**Attack Results of CelebA.** The limited effectiveness of these defenses on CelebA is illustrated in Tab. 4. In comparison to CIFAR-10, CelebA shows a more robust performance in terms of test accuracy. This is because CelebA is employed for



a simpler binary classification task (smile classification), making the model more easily convergent even in the presence of defense methods.

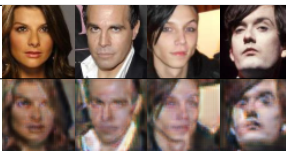



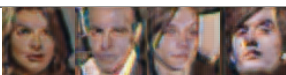

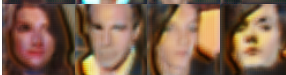

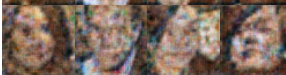
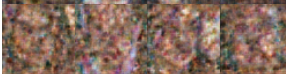
Table 3. Results on CIFAR-10 at layer 2 with smaller  $\epsilon$ .

DP ( $\epsilon$ )	Test Acc (%)	SSIM $\uparrow$	PSNR $\uparrow$	LPIPS $\downarrow$	
6	61.61%	0.590	17.49	0.496	
1	61.17%	0.582	17.43	0.522	

**Results with Smaller  $\epsilon$  on CIFAR-10.** Table 3 presents the results with smaller  $\epsilon$  on CIFAR-10. We observe an interesting phenomenon: as the applied noise increased, there is a nonlinear relationship with the defense results. The possible reason is that the noise can only act on partial gradients (client), limiting its effectiveness.

**Possible Adaptive Defenses.** We discuss two potential adaptive defenses. One is that the client adopts an adversarial loss to enhance robustness against DRA [11, 21]. Though adversarial learning proves effective against certain known attacks, client should carefully consider the additional training overhead and utility degradation it introduces. Another promising approach is to craft noise against FORA to increase the inconsistency between client and substitute client in feature space, which would make attack more difficult [20].

Table 4. Effect of utility and FORA performance against distance correlation minimization, differential privacy and noise obfuscation on CelebA at layer 2.

Defense Hyperparam	Test Acc (%)	SSIM $\uparrow$	PSNR $\uparrow$	LPIPS $\downarrow$	
0 (w/o defense)	91.91	0.476	17.11	0.381	
<b>DCOR (<math>\alpha</math>)</b>					
0.2	91.96	0.407	15.46	0.470	
0.5	92.24	0.355	14.58	0.428	
0.8	91.93	0.266	12.03	0.635	
<b>DP (<math>\epsilon</math>)</b>					
$+\infty$	91.91	0.460	16.21	0.424	
100	91.00	0.460	16.65	0.507	
10	91.41	0.435	16.54	0.455	
<b>NO (<math>\sigma</math>)</b>					
0.5	92.19	0.378	16.63	0.568	
1.0	92.35	0.311	15.31	0.666	
2.0	92.40	0.140	12.61	0.704	

## D. Limitation and Future Work

In this section, we discuss the limitations of our proposed method FORA and some possible enhancements for future work. Previous work and FORA lack sufficient experiments on larger models and datasets *e.g.* vision transformer [3], so we encourage future work to pay more attention on larger models. Additionally, although FORA only requires auxiliary data of the same type to launch an attack, exploring how to reconstruct client inputs in more challenging scenarios, such as when attackers do not know the data type, is also unsolved. We hope our work can contribute to better exploring vulnerabilities of SL and raising awareness of privacy issues within the community.

## E. Label-Protected SL

In label-protected SL [17] shown in Fig. 8, besides the client model and server model, there is also a portion of model called the top model retained on the client. In this scenario, labels are treated as private information and kept locally on the client. Differing from label-share SL, the server model's results are forwarded to the top model for further forward propagation. The top model then calculates the loss using labels and received results, transferring relevant gradients to the server for parameter updates.

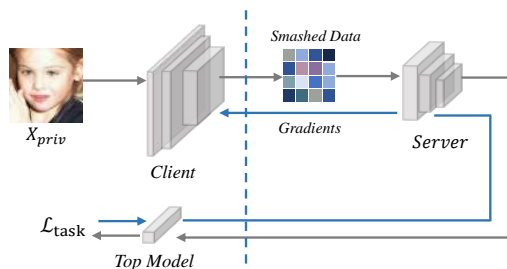


Figure 8. Architecture of label-protected split learning.

## References

- [1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016. 4
- [2] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018. 1
- [3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 6
- [4] Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33*, pages 1–12. Springer, 2006. 4
- [5] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014. 4
- [6] Jiayun Fu, Xiaojing Ma, Bin B. Zhu, Pingyi Hu, Ruixin Zhao, Yaru Jia, Peng Xu, Hai Jin, , and Dongmei Zhang. Focusing on pinocchio’s nose: A gradients scrutinizer to thwart split-learning hijacking attacks using intrinsic attributes. In *30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27-March 3, 2023*. The Internet Society, 2023. 4
- [7] Xinben Gao and Lan Zhang. PCAT: Functionality and data stealing from split learning by Pseudo-Client attack. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5271–5288, Anaheim, CA, 2023. USENIX Association. 4
- [8] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 1
- [9] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1
- [10] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. IEEE, 2019. 4
- [11] Jingtao Li, Adnan Siraj Rakin, Xing Chen, Zhezhi He, Deliang Fan, and Chaitali Chakrabarti. Ressfl: A resistance transfer framework for defending model inversion attack in split federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10194–10202, 2022. 5
- [12] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015. 1
- [13] Dario Pasquini, Giuseppe Ateniese, and Massimo Bernaschi. Unleashing the tiger: Inference attacks on split learning. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 2113–2129, 2021. 4
- [14] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015. 4
- [15] Gábor J Székely, Maria L Rizzo, and Nail K Bakirov. Measuring and testing dependence by correlation of distances. 2007. 4
- [16] Tom Titcombe, Adam J Hall, Pavlos Papadopoulos, and Daniele Romanini. Practical defences against model inversion attacks for split neural networks. *arXiv preprint arXiv:2104.05743*, 2021. 4
- [17] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564*, 2018. 6
- [18] Praneeth Vepakomma, Otkrist Gupta, Abhimanyu Dubey, and Ramesh Raskar. Reducing leakage in distributed deep learning for sensitive health data. *arXiv preprint arXiv:1812.00564*, 2, 2019. 4
- [19] Praneeth Vepakomma, Abhishek Singh, Otkrist Gupta, and Ramesh Raskar. Nopeek: Information leakage reduction to share activations in distributed deep learning. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pages 933–942. IEEE, 2020. 4
- [20] Zhibo Wang, He Wang, Shuaifan Jin, Wenwen Zhang, Jiahui Hu, Yan Wang, Peng Sun, Wei Yuan, Kaixin Liu, and Kui Ren. Privacy-preserving adversarial facial features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8212–8221, 2023. 5
- [21] Taihong Xiao, Yi-Hsuan Tsai, Kihyuk Sohn, Manmohan Chandraker, and Ming-Hsuan Yang. Adversarial learning of privacy-preserving and task-oriented representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12434–12441, 2020. 5