Differentially Private CutMix for Split Learning with Vision Transformer

Seungeun Oh, Sihun Baek, Hyelin Nam, Seong-Lyun Kim[*]
Yonsei University
{seoh,shbaek,hlnam,slkim}
@ramo.yonsei.ac.kr

Jihong Park
Deakin University
jihong.park
@deakin.edu.au

Praneeth Vepakomma, Ramesh Raskar
Massachusetts Institute of Technology
{vepakom,raskar}
@mit.edu

Mehdi Bennis
University of Oulu
mehdi.bennis
@oulu.fi

Abstract

Recently, vision transformer (ViT) has started to outpace the conventional CNN in computer vision tasks. Considering privacy-preserving distributed learning with ViT, federated learning (FL) communicates models, which becomes ill-suited due to ViT's large model size and computing costs. Split learning (SL) detours this by communicating smashed data at a cut-layer, yet suffers from data privacy leakage and large communication costs caused by high similarity between ViT's smashed data and input data. Motivated by this problem, we propose *DP-CutMixSL*, a differentially private (DP) SL framework by developing *DP patch-level randomized CutMix (DP-CutMix)*, a novel privacy-preserving inter-client interpolation scheme that replaces randomly selected patches in smashed data. By experiment, we show that DP-CutMixSL not only boosts privacy guarantees and communication efficiency, but also achieves higher accuracy than its Vanilla SL counterpart. Theoretically, we analyze that DP-CutMix amplifies Rényi DP (RDP), which is upper-bounded by its Vanilla Mixup counterpart.

1 Introduction

Motivation: Privacy-Preserving Distributed ML for ViT Edge devices such as phones, cameras, and e-health wearables generate the sheer amount of fresh data [1]. To exploit these user data for machine learning (ML) without violating data privacy, federated learning (FL) is gaining increasing attention, which keeps raw data locally stored while only exchanging and averaging model parameters across devices [2, 3]. In particular, FL has been notably successful in computer vision tasks with the *de facto* standard convolutional neural network (CNN) architectures [4, 5]. However, recently vision transformer (ViT) has been aggressively taking over the throne of CNN [6], questioning the effectiveness of FL. In fact, ViT is often larger than CNN, and this bodes ill for FL by imposing excessive energy and communication burdens on devices [7, 8]. Alternatively, split learning (SL) can cope with large models via model partitioning [9, 10]. In SL, each device locally stores only a tiny fraction of the entire model, and offloads the rest to a parameter server, between which devices exchange their cut-layer forward activations with the server, referred to as *smashed data*. Notwithstanding, ViT commonly lacks pooling and convolutional layers [6, 11, 12], making smashed data similar to their raw data as visualized in Fig. 4 of Appendix A. This may entail huge costs and privacy leakage as opposed to its counterpart SL with CNN [13, 14, 15].

[*] Corresponding author.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

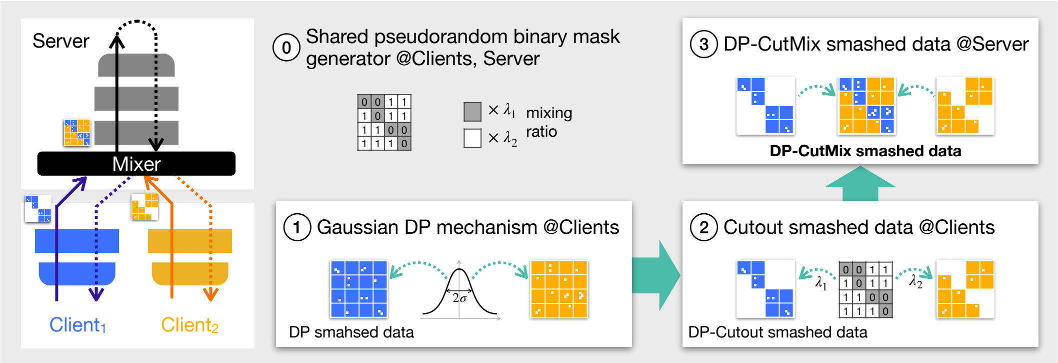

Server
Mixer
Client₁
Client₂
0 Shared pseudorandom binary mask generator @Clients, Server
0 0 1 1
1 0 1 1
1 1 0 0
1 1 1 0
× λ_1 mixing ratio
× λ_2
3 DP-CutMix smashed data @Server
DP-CutMix smashed data
1 Gaussian DP mechanism @Clients
2σ
DP smahsed data
2 Cutout smashed data @Clients
λ_1
λ_2
DP-Cutout smashed data

Figure 1: A schematic illustration of DP-CutMixSL operations.

Contributions: DP-CutMixSL To address the aforementioned issues, inspired from the patchfied smashed data in ViT [6] and the CutMix technique [16], we propose *DP-CutMixSL* [2], a differentially private (DP) SL framework with ViT via patch-level randomized CutMix. As Fig. 1 demonstrates, following the Gaussian DP mechanism [17, 18, 19], each device in DP-CutMixSL first injects random Gaussian noise into smashed data, followed by punching randomly selected patches, yielding *Cutout smashed data* as analogous to those of Cutout [20]. These Cutout smashed data are uploaded to and put together by the server, resulting in *DP-CutMix smashed data* that continue feed-forward propagation. Compared to SL with the Gaussian DP mechanism (*DP-SL*), we theoretically prove that the proposed randomized CutMix operation in DP-CutMixSL amplifies the DP guarantee of smashed data, by up to its upper-bound baseline *DP-MixSL* obtained by replacing CutMix with Mixup [21] that simply superimposes the entire patches from each of different smashed data. By experiment, we show that DP-CutMixSL achieves the highest accuracy, followed by DP-MixSL and DP-SL. It is worth noting that while most of the existing works apply Cutout and CutMix at pixel levels for intra-dataset interpolations [22, 23], we utilize them at patch levels for privacy-preserving inter-dataset interpolations across different devices, i.e., privacy-preserving distributed ML.

2 DP-CutMixSL: Patch-Level Randomized CutMix Operations for ViT

The major difference between ViT and CNN can be summarized as follows: i) As shown in Fig. 4 of Appendix A, ViT has less feature distortion for the input data of the hidden representation (i.e. smashed data) due to the absence of a pooling layer, ii) Due to its own self-attention mechanism driven by embedding process, ViT captures global spatial information whereas CNN focuses on local spatial information, iii) The above operations of ViT run at patch-level.

At first, i) implies that regularization of the hidden representation in ViT is as efficient as in the input data. Conversely, the mutual information about the input data of the hidden representation is high, leading to data privacy leakage. Next, due to the property of ViT to learn global spatial information mentioned in ii), ViT has more robustness to large-scale noise applied to the fraction of the image [25], thereby it is suitable for Cutout [20] or CutMix regularization. Finally, iii) suggests the possibility of a patch-scale regularizer. Integrating the above yields a common solution, patch-level randomized CutMix of hidden representations, short for *patch CutMix*.

Let i and $\mathbf{C}$ be a subscript for a client and a set of clients, respectively. As observed in Fig. 1, a mixer, which may be a third-party entity, first generates random sequences M_i with the mixing ratio $\lambda_i \in [0, 1]\ \forall i \in \mathbf{C}$ following a Dirichlet multinomial distribution [26], where $\sum_i \lambda_i = 1$. For instance, if a smashed data s_i consists of N patches, M_i is a random binary sequence of length N to control the on-off of each patch, and its non-zero element is $\lceil \lambda_i \cdot N \rceil$.

Then, the i-th client acquires the Cutout smashed data $\bar{s}_i$ by masking the smashed data ($\bar{s}_i = M_i \odot s_i$), obtained by passing the input data through the lower model segment, via the random sequence downloaded from the mixer. When the Cutout smashed data and its label are uploaded to the server,

[2] An early version of this work was presented at FL-IJCAI 2022 [24]. Compared to [24] proposing CutMixSL and focusing its communication efficiency, this work proposes DP-CutMixSL while studying its DP analysis and the privacy-accuracy trade-off.

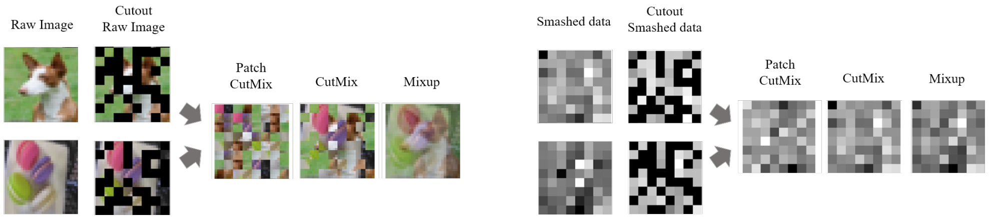
Algorithm 1 DP-CutMixSL

```
requirements: $w = [w_{c,i}, w_s]^T$ ($w_{c,i}$: lower model segment, $w_s$: upper model segment)
              $\eta$: learning rate
while $w$ not converged do
    /*Runs on mixer*/
    samples $\{a_1, .., a_n\} \sim \text{Dir}(\bar{\alpha})$
    generates pseudo random sequences $M_i$ for all $i$    ▷ Pseudorandom binary mask generation
    unicasts $M_i$ to $i$-th client for all $i$

    /*Runs on client $i \in$ C*/
    generates smashed data $s_i$ by passing input data $x_i$ through $w_{c,i}$
    produces $\bar{s}_i$ by masking $s_i$ via $M_i$    ▷ Cutout smashed data
    produces $\bar{s}_i'$ by applying Gaussian mechanism    ▷ DP-Cutout smashed data
    uploads $\bar{s}_i'$ to the server

    /*Runs on server*/
    produces $\tilde{s}_i'$ via $\bar{s}_i'$ aggregation for all $i$    ▷ DP-CutMix smashed data
    generates loss $\sum_i L_i$ by passing $\tilde{s}_i'$ through $w_s$ in parallel
    updates $w_s$ via $w_s \leftarrow w_s - \eta \cdot \nabla_{w_s}(\sum_i L_i)$    ▷ Upper model segment update
    unicasts $i$-th cut-layer gradient to $i$-th client for all $i$

    /*Runs on client $i \in$ C*/
    updates $w_{c,i}$ via $w_{c,i} \leftarrow w_{c,i} - \eta \cdot \nabla_{w_{c,i}}(\sum_i L_i)$    ▷ Lower model segment update
end while
```



(a) Raw images.

(b) Smashed data.

Figure 2: Examples of data obtained by performing various interpolation schemes on (a) raw image and (b) smashed data.

we assume that a gaussian mechanism is applied to them, generating the following DP-Cutout smashed data and label containing white gaussian noise of N_s and N_y, respectively:

$$\bar{s}_i' = \bar{s}_i + N_s = M_i \odot s_i + N_s, \tag{1}$$

$$\bar{y}_i' = \bar{y}_i + N_y. \tag{2}$$

The server aggregates DP-Cutout smashed data from all clients and generates *DP-CutMix smashed data* in the following way:

$$\tilde{s}_{i,j}' = \bar{s}_i' + \bar{s}_j', \quad \tilde{y}_{i,j}' = \lambda_i \cdot \bar{y}_i' + \lambda_j \cdot \bar{y}_j', \quad \text{for } j \neq i. \tag{3}$$

Next, the rest of DP-CutMixSL's operation, equal to that of Vanilla SL, performing FP & BP on the server-side model follows. The said operation of DP-CutMixSL is detailed by the pseudo code of Algorithm 1. Fig. 2 also provides image samples of smashed data as well as input data to which the proposed patch CutMix is applied compared to those of Mixup and Vanilla CutMix.

As a result, DP-CutMixSL can benefit both in terms of privacy leakage and communication cost, in a way that only fraction of the smashed data is shared to the server, even ejected with gaussian noise. Note that random sequences used for smashed data masking are mutually exclusive and collectively exhaustive at the patch-level, so that there are no blank patches in DP-CutMix smashed data.

3 DP-CutMixSL: Differential Privacy Analysis

Let $\mathcal{D} = \{(s_1, y_1), .., (s_n, y_n)\}$ be a set consisting of n clients' pairs of smashed data $s_i \in \mathbb{R}^{N \times P^2 \times C}$ and the corresponding label $y \in \mathbb{R}^L$ is a one-hot encoding vector, where P denotes the size of patch, respectively. We assume that the each element of the smashed data and ground-truth label is upper bounded as follows: $s_i \in [0, \Delta]^{D_s}$ and $y_i \in [0, 1]^{D_y}$, where $D_s = NP^2C$ and $D_y = L$. In addition, λ_i is the mixing ratio of the i-th client, and N_s and N_y are white gaussian noise with dimensions D_s and D_y, respectively, i.e., $N_s \sim N(0, \sigma_s^2 I_{D_s})$ and $N_y \sim N(0, \sigma_y^2 I_{D_y})$ for some (σ_s, σ_y). Then, we derive the *Rényi differential privacy (RDP)* [19] of the proposed DP-CutMixSL, which is compared with those of DP-SL and DP-MixSL as follows.

Theorem 1. For a given order α, the RDP privacy budgets $\epsilon_o(\alpha)$, $\epsilon_{Mix}(\alpha)$, and $\epsilon_{CutMix}(\alpha)$ of DP-SL, DP-MixSL and DP-CutMixSL satisfy $\epsilon_{Mix}(\alpha) \leq \epsilon_{CutMix}(\alpha) \leq \epsilon_o(\alpha)$ where:

$$\epsilon_o(\alpha) = \frac{\alpha}{2}\left(\frac{\Delta^2 D_s}{\sigma_s^2} + \frac{D_y}{\sigma_y^2}\right), \tag{4}$$

$$\epsilon_{Mix}(\alpha) = \frac{\alpha\left(\max_{i \in \mathbf{C}} \lambda_i\right)^2}{2}\left(\frac{\Delta^2 D_s}{\sigma_s^2} + \frac{D_y}{\sigma_y^2}\right), \tag{5}$$

$$\epsilon_{CutMix}(\alpha) = \frac{\alpha\left(\max_{i \in \mathbf{C}} \lambda_i\right)}{2}\left(\frac{\Delta^2 D_s}{\sigma_s^2} + \frac{D_y\left(\max_{i \in \mathbf{C}} \lambda_i\right)}{\sigma_y^2}\right). \tag{6}$$

Sketch of Proof. For each technique, we derive its output representation, followed by calculating the RDP bound using the output via the Rényi divergence formula for a multivariate Gaussian distribution [18]. Applying this to both the smashed data and the label and combining them via the sequential composition rule completes the proof. The details are deferred to Appendix B. ■

Since $\lambda_i \in [0, 1]$, DP-MixSL achieves the highest RDP guarantee (i.e., tightest RDP bound) compared to DP-CutMixSL and DP-SL, with the help of the inherent distortion property of interpolations [27, 28, 29]. It is worth noting that the case only when $\max_{i \in \mathbf{C}} \lambda_i = 1$, i.e., a single client scenario with $|\mathbf{C}| = 1$, the equality conditions $\epsilon_{Mix}(\alpha) = \epsilon_{CutMix}(\alpha) = \epsilon_o(\alpha)$ hold. In other words, none of equality does not hold for multi clients. Note here that we focus only on the RDP guarantees of smashed data. Smashed data are vulnerable to reconstruction attacks [30], threatening the privacy of raw data, which is discussed in Appendix C. In addition, while we focus on the label privacy in the FP, the label privacy can also be leaked from gradients in the BP via white-box attacks. To prevent this, BP label privacy guaranteeing method such as GradPerturb [31] can additionally be integrated, which is deferred to future research.

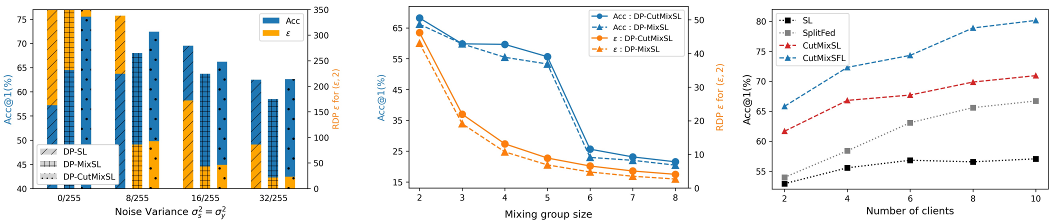
4 Numerical Evaluation

In this section, we measure the accuracy, RDP bound (ϵ), and scalability of DP-CutMixSL compared to those of SplitFed [14], DP-SL, DP-MixSL, and etc. In Table 1, both the CIFAR-10 dataset [32] and the Fashion-MNIST dataset [33] are utilized under three types of models: ViT-tiny [34], PiT-tiny [35], and VGG-16 [36]. Here, PiT is a transformer structure equipped with a pooling layer and is a model between ViT and CNN. For all SL algorithms, we assume that the cut-layer is located after embedding process. Other parameters especially for RDP calculation are as follows: patch size $N = 64$, $D_s = 20$, $D_y = 10$, $\Delta = 0.2$, $\lambda_i = 1/n\ \forall i$ (uniform), and RDP parameter $\alpha = 2$.

Table 1 shows the top-1 accuracy for several SL methods including the proposed CutMixSL in a noiseless environment. As seen at Table 1, except for one case, where SL w. Mixup is used with CIFAR-10 dataset and VGG-16 model, the CutMixSL outperforms other state-of-the-art SL algorithms in terms of top-1 accuracy. This is rooted in the difference between ViT and CNN mentioned in Sec. 2. When learning spatial information, ViT focuses on globality due to the self-attention mechanism, whereas CNN focuses on locality. Hence, a patch CutMix in which certain patches are replaced by patches of other smashed data may cause significant information loss in CNN. On the other hand, interpolation such as mixup is less likely to yield large information loss, thereby CNN and ViT are suitable for mixup and patch CutMix, respectively. Comparing CutMixSL and SL w. Vanilla CutMix in Table 1 proves that patch-level random punching is more efficient than bounding box-based process. This is because the random punching method is a regularizer well suited to ViT, where all operations operate at patch-level, unlike Vanilla CutMix's bounding box where the size is regardless of patch size. From a dropout [37] perspective, the random punching method of CutMixSL is more similar to dropout than mixup or Vanilla CutMix, leading to accuracy gains.

Table 1: Top-1 accuracy of SL-based techniques w.r.t various model types and datasets.

Method ($\|\mathbf{C} = 10\|$)	Models w/ CIFAR-10			Models w/ Fashion-MNIST		
	ViT-Tiny	PiT-Tiny	VGG-16	ViT-Tiny	PiT-Tiny	VGG-16
Standalone	48.84	47.77	54.97	77.65	78.21	80.12
SL [10]	57.21	52.28	62.62	85.68	82.35	84.39
SplitFed [14]	67.88	55.63	63.98	89.17	84.27	87.34
Standalone w. Cutout [20]	53.86	50.28	56.65	88.46	86.48	88.17
SL w. Mixup	69.23	64.89	**68.20**	88.21	87.62	88.53
SL w. Vanilla CutMix	71.78	58.21	33.50	87.86	86.31	89.01
CutMixSL (proposed)	**73.77**	**71.26**	67.53	**89.75**	**89.25**	**89.45**



(a) Acc-ϵ per noise variance.

(b) Acc-ϵ of mixing methods.

(c) Scalability.

Figure 3: Accuracy and ϵ under the CIFAR-10 dataset: (a) accuracy and ϵ of DP-SL, DP-MixSL, and DP-CutMixSL w.r.t noise variance; (b) accuracy and ϵ of DP-MixSL and DP-CutMixSL w.r.t the mixing group size; (c) accuracy of various SL-based techniques according to number of clients.

Fig. 3a shows the effect of noise variance on accuracy and ϵ. In terms of accuracy, DP-CutMixSL is the best, except when the noise variance is $16/255$. Compared to DP-MixSL, DP-CutMixSL has superior performance in all cases. Looking at ϵ, however, DP-CutMixSL has a tighter RDP bound compared to DP-SL, but has a larger ϵ in comparison with DP-MixSL, showing the *accuracy-privacy trade-off*. In Fig. 3b, the size of a mixing group, a set of clients taking a mixup or CutMix, varies both in DP-CutMixSL and DP-MixSL. In both DP-CutMixSL and DP-MixSL, the accuracy and ϵ decrease as the mixing group size increases, also resulting in the accuracy-privacy trade-off. The decrease in ϵ according to the mixing group size can be explained by the "Hiding in the crowd" effect [38], and the decrease in accuracy is also explained in connection with the information loss mentioned above. Fig. 3c shows the curve of scalability, in terms of accuracy increase according to the number of clients. In Fig. 3c, all SL techniques including CutMixSL guarantee scalability when the client increases from 2 to 10, and the accuracy of CutMixSFL, which introduced SplitFed's weight averaging of lower model segment to CutMixSL, is further improved.

5 Concluding Remarks

In this work, we proposed DP-CutMixSL for privacy-preserving distributed ML for ViT by exploiting CutMix for inter-dataset interpolations. We theoretically analyzed its DP guarantee, and numerically showed its achieving the highest accuracy compared to two baselines, DP-SL and DP-MixSL. While we focus only on generating a single CutMix output for two or more inputs, it is possible to generate multiple outputs by changing the mixing ratio for interpolations with finer resolution as done in [39], which could be an interesting topic for future study. Furthermore, based on the simulation results showing the effectiveness of the proposed method even in the presence of several pooling and convolutional layers, it is worth investigating other patchified architectures in different domains for future research.

Acknowledgments and Disclosure of Funding

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-02208, Sub-THz Augmented Routing and Transmission for 6G).Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-02208, Sub-THz Augmented Routing and Transmission for 6G).

References

[1] Jihong Park, Sumudu Samarakoon, Mehdi Bennis, and Mérouane Debbah. Wireless network intelligence at the edge. *Proceedings of the IEEE*, 107(11):2204–2239, 2019.

[2] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[3] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[4] Yang Liu, Anbu Huang, Yun Luo, He Huang, Youzhi Liu, Yuanyuan Chen, Lican Feng, Tianjian Chen, Han Yu, and Qiang Yang. Fedvision: An online visual object detection platform powered by federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13172–13179, 2020.

[5] Chaoyang He, Alay Dilipbhai Shah, Zhenheng Tang, Di Fan1Adarshan Naiynar Sivashunmugam, Keerti Bhogaraju, Mita Shimpi, Li Shen, Xiaowen Chu, Mahdi Soltanolkotabi, and Salman Avestimehr. Fedcv: a federated learning framework for diverse computer vision tasks. *arXiv preprint arXiv:2111.11066*, 2021.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[7] 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.

[8] Abhishek Singh, Praneeth Vepakomma, Otkrist Gupta, and Ramesh Raskar. Detailed comparison of communication efficiency of split learning and federated learning. *arXiv preprint arXiv:1909.09145*, 2019.

[9] Otkrist Gupta and Ramesh Raskar. Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications*, 116:1–8, 2018.

[10] 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.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[13] Chandra Thapa, Mahawaga Arachchige Pathum Chamikara, and Seyit Camtepe. Splitfed: When federated learning meets split learning. *CoRR*, abs/2004.12088, 2020.

[14] Chandra Thapa, Mahawaga Arachchige Pathum Chamikara, Seyit Camtepe, and Lichao Sun. Splitfed: When federated learning meets split learning. *arXiv preprint arXiv:2004.12088*, 2020.

[15] Yansong Gao, Minki Kim, Chandra Thapa, Sharif Abuadbba, Zhi Zhang, Seyit Camtepe, Hyoungshick Kim, and Surya Nepal. Evaluation and optimization of distributed machine learning techniques for internet of things. *IEEE Transactions on Computers*, 2021.

[16] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019.

[17] 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.

[18] Manuel Gil, Fady Alajaji, and Tamas Linder. Rényi divergence measures for commonly used univariate continuous distributions. *Information Sciences*, 249:124–131, 2013.

[19] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275. IEEE, 2017.

[20] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[21] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[22] Caidan Zhao and Yang Lei. Intra-class cutmix for unbalanced data augmentation. In *2021 13th International Conference on Machine Learning and Computing*, pages 246–251, 2021.

[23] Lianbo Zhang, Shaoli Huang, and Wei Liu. Intra-class part swapping for fine-grained image classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3209–3218, 2021.

[24] Sihun Baek, Jihong Park, Praneeth Vepakomma, Ramesh Raskar, Mehdi Bennis, and Seong-Lyun Kim. Visual transformer meets cutmix for improved accuracy, communication efficiency, and data privacy in split learning. *arXiv preprint arXiv:2207.00234*, 2022.

[25] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34, 2021.

[26] Yvonne M Bishop, Stephen E Fienberg, and Paul W Holland. *Discrete multivariate analysis: theory and practice*. Springer Science & Business Media, 2007.

[27] Yusuke Koda, Jihong Park, Mehdi Bennis, Praneeth Vepakomma, and Ramesh Raskar. Airmixml: Over-the-air data mixup for inherently privacy-preserving edge machine learning. *arXiv preprint arXiv:2105.00395*, 2021.

[28] Eitan Borgnia, Jonas Geiping, Valeriia Cherepanova, Liam Fowl, Arjun Gupta, Amin Ghiasi, Furong Huang, Micah Goldblum, and Tom Goldstein. Dp-instahide: Provably defusing poisoning and backdoor attacks with differentially private data augmentations. *arXiv preprint arXiv:2103.02079*, 2021.

[29] Kangwook Lee, Hoon Kim, Kyungmin Lee, Changho Suh, and Kannan Ramchandran. Synthesizing differentially private datasets using random mixing. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 542–546. IEEE, 2019.

[30] Grzegorz Gawron and Philip Stubbings. Feature space hijacking attacks against differentially private split learning. *arXiv preprint arXiv:2201.04018*, 2022.

[31] Xin Yang, Jiankai Sun, Yuanshun Yao, Junyuan Xie, and Chong Wang. Differentially private label protection in split learning. *arXiv preprint arXiv:2203.02073*, 2022.

[32] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[33] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[34] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[35] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. *CoRR*, abs/2103.16302, 2021.

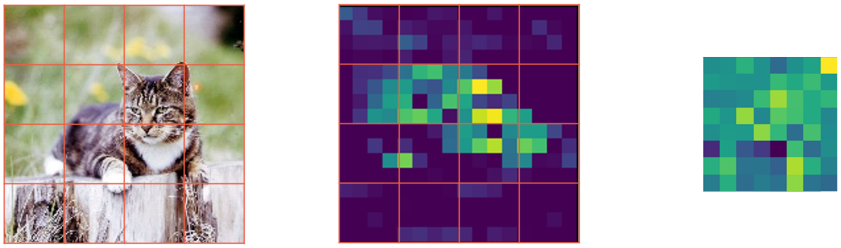
[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[37] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[38] Eunjeong Jeong, Seungeun Oh, Jihong Park, Hyesung Kim, Mehdi Bennis, and Seong-Lyun Kim. Hiding in the crowd: Federated data augmentation for on-device learning. *IEEE Intelligent Systems*, 36(5):80–87, 2020.

[39] Seungeun Oh, Jihong Park, Praneeth Vepakomma, Sihun Baek, Ramesh Raskar, Mehdi Bennis, and Seong-Lyun Kim. Locfedmix-sl: Localize, federate, and mix for improved scalability, convergence, and latency in split learning. In *Proceedings of the ACM Web Conference 2022*, pages 3347–3357, 2022.

A Image Visualization in CNN and ViT



(a) Input raw image. (b) Smashed data of ViT. (c) Smashed data of CNN.

Figure 4: Comparison between raw image and smashed data of ViT and CNN.

B RDP Analysis

Before going further, the definition of RDP is as follows:

Definition B.1 ((α, ϵ)-RDP [19]). A randomized mechanism $f : \mathcal{D} \rightarrow \mathcal{R}$ is said to have ϵ-Rényi differential privacy of order α, or (α, ϵ)-RDP for short, if for any adjacent $D, D' \in \mathcal{D}$ it holds that

$$D_\alpha(f(D)||f(D')) \leq \epsilon \tag{7}$$

A strong privacy guarantee implies that one cannot distinguish whether D or D' was used to produce an outcome of mechanism.

B.1 DP-SL

Here, our baseline DP mechanism is Gaussian mechanism. Then, the output of DP-SL, which are the smashed data and its label to which gaussian noise is applied, respectively, is as follows:

$$s_i' = s_i + N_s, \tag{8}$$

$$y_i' = y_i + N_y, \tag{9}$$

where $N_s \sim N(0, \sigma_s^2 I_{D_s})$ and $N_y \sim N(0, \sigma_y^2 I_{D_y})$ for some (σ_s, σ_y).

By using the Definition B.1 and Rényi divergence formula from [18], the RDP bound of gaussian mechanism $\mathcal{M}$ for DP-SL, $\epsilon_o(\alpha)$ can be expressed as:

$$\epsilon_o(\alpha) = \sup_{\mathcal{D},\mathcal{D}'} D_\alpha(\mathcal{M}(\mathcal{D})||\mathcal{M}(\mathcal{D}')) = \sup_{\mathcal{D},\mathcal{D}'} \frac{\alpha}{2\sigma^2}||\mu_X^{\mathcal{D}} - \mu_X^{\mathcal{D}'}||^2, \tag{10}$$

where $\mathcal{M}(\mathcal{D}) \sim N(\mu_X^{\mathcal{D}}, \sigma_X^2)$ and $\mathcal{M}(\mathcal{D}') \sim N(\mu_X^{\mathcal{D}'}, \sigma_X^2)$.

Let $s_{(i,k)}^{\mathcal{D}}$ denote the k-th element of smashed data $s_i^{\mathcal{D}}$ in dataset $\mathcal{D}$, where $s_i^{\mathcal{D}} = [s_{(i,1)}^{\mathcal{D}}, ..., s_{(i,D_s)}^{\mathcal{D}}]$. Then, $s_i'^{\mathcal{D}}$ obtained by passing $s_i^{\mathcal{D}}$ through (8) follows a gaussian distribution with mean $s_i^{\mathcal{D}}$ and variance σ_s^2 (In element perspective, $s_{(i,k)}'^{\mathcal{D}} \sim N(s_{(i,k)}^{\mathcal{D}}, \sigma_s^2)$ for $k \in [D_s]$).

For two sets of smashed data $\mathcal{D}$ and $\mathcal{D}'$ where only the i'-th smashed data is different, (10) becomes:

$$\sup_{\mathcal{D},\mathcal{D}'} \frac{\alpha}{2\sigma_s^2}||\mu_X^{\mathcal{D}} - \mu_X^{\mathcal{D}'}||^2 = \frac{\alpha}{2\sigma_s^2}\sum_{k=1}^{D_s}(s_{(i',k)}^{\mathcal{D}} - s_{(i',k)}^{\mathcal{D}'})^2. \tag{11}$$

Considering the element-wise upper bound of DP-SL's smashed data yields the following formula:

$$\sum_{k=1}^{D_s}(s_{(i',k)}^{\mathcal{D}} - s_{(i',k)}^{\mathcal{D}'})^2 \leq \Delta^2 \cdot D_s. \tag{12}$$

Therefore, gaussian mechanism for smashed data in DP-SL is (α, ϵ_o)-RDP, where

$$\epsilon_o(\alpha) = \alpha \frac{\Delta^2 \cdot D_s}{2\sigma_s^2}. \tag{13}$$

Likewise, RDP bound of groud-truth label in DP-SL can be calculated, yielding $\epsilon_o(\alpha)$ as below:

$$\epsilon_o(\alpha) = \frac{\alpha}{2}\bigg(\frac{\Delta^2 D_s}{\sigma_s^2} + \frac{D_y}{\sigma_y^2}\bigg). \tag{14}$$

B.2 DP-MixSL

The output of DP-MixSL can be expressed as $\hat{s} = \sum_{i=1}^{n} \lambda_i s_i'$, where $\{\lambda_1, ..., \lambda_n\}$ is a set of mixing ratio of each client's smashed data. Also, DP-MixSL executes mixup operation similar to its corresponding label.

Then, for two adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$, (10) becomes:

$$\sup_{\mathcal{D},\mathcal{D}'} \frac{\alpha}{2\sigma_s^2} \|\mu_X^{\mathcal{D}} - \mu_X^{\mathcal{D}'}\|^2 = \frac{\alpha}{2\sigma_s^2} \sum_{k=1}^{D_s} \big(\lambda_{i'}(s_{(i',k)}^{\mathcal{D}} - s_{(i',k)}^{\mathcal{D}'})\big)^2. \tag{15}$$

This bound can be maximized when the i'-th mixing ratio is the largest value of λ_i for all $i \in \mathbf{C}$. In this case, from 15, we have

$$\sum_{k=1}^{D_s} \big(\lambda_{i'}(s_{(i',k)}^{\mathcal{D}} - s_{(i',k)}^{\mathcal{D}'})\big)^2 \leq (\max_{i\in\mathbf{C}} \lambda_i)^2 \Delta^2 D_s. \tag{16}$$

By applying the same process above to the label, we have

$$\epsilon_{Mix}(\alpha) = \epsilon_o(\alpha) \cdot \bigg(\max_{i\in\mathbf{C}} \lambda_i\bigg)^2 \tag{17}$$

$$= \frac{\alpha \left(\max_{i\in\mathbf{C}} \lambda_i\right)^2}{2} \bigg(\frac{\Delta^2 D_s}{\sigma_s^2} + \frac{D_y}{\sigma_y^2}\bigg). \tag{18}$$

B.3 DP-CutMixSL

The output of DP-CutMixSL can be represented by $\tilde{s} = \sum_{i=1}^{n} M_i \odot s_i'$, whereas its operation on label is the same as in DP-MixSL. For two adjacent datasets with only one smashed data different, DP-CutMixSL only needs to calculate bounds only for the element in which the smashed data is masked, in contrast to DP-MixSL, where the smashed data is melted in the whole element.

That is, assuming that the number of 1 elements included in the i'-th mask is $N_{i'}$, the following inequality is derived from (10):

$$\sup_{\mathcal{D},\mathcal{D}'} \frac{\alpha}{2\sigma_s^2} \|\mu_X^{\mathcal{D}} - \mu_X^{\mathcal{D}'}\|^2 \leq \frac{\alpha}{2\sigma_s^2} N_{i'} \Delta^2 = \frac{\alpha}{2\sigma_s^2} \lambda_{i'} D_s \Delta^2. \tag{19}$$

(19) has an upper bound as shown below when $\lambda_{i'}$ is the maximum among λ_i $\forall i$:

$$\frac{\alpha}{2\sigma_s^2} \lambda_{i'} D_s \Delta^2 \leq (\max_{i\in\mathbf{C}} \lambda_i) \Delta^2 D_s. \tag{20}$$

Then, we have $\epsilon_{CutMix}(\alpha)$ for DP-CutMixSL mechanism as follows:

$$\epsilon_{CutMix}(\alpha) = \frac{\alpha(\max_{i\in\mathbf{C}} \lambda_i)}{2} \bigg(\frac{\Delta^2 D_s}{\sigma_s^2} + \frac{(\max_{i\in\mathbf{C}} \lambda_i) D_y}{\sigma_y^2}\bigg). \tag{21}$$

Table 2: Privacy leakage measured by the reconstruction loss (MSE).

Type	Train Dataset (10%)	Train Dataset (100%)
Smashed data	0.0091	0.0056
Cutout	**0.0920**	**0.0829**
Mixup	0.0402	0.0351
Patch CutMix	0.0458	0.0434

C Robustness to Reconstruction Attack

Table 2 shows loss between raw data and reconstructed data generated from different types of mixing methods or datasets. To restore raw data from reconstructed data, we utilize a decoder model, comprised of two convolutional layers with additional interpolation methods to adaptively match the dimension to the aimed data size. For comparison, we train the decoder model with training datasets of two different sizes.

As a result, regardless of the training dataset size, the reconstruction loss was large in the following order: Cutout, patch CutMix, Mixup, and smashed data. In other words, it has robustness against reconstruction attacks in that order. Except for smashed data, Mixup is most vulnerable to reconstruction attack, because information leakage occurs over the entire area even though linear interpolation is taken. The proposed patch CutMix has relatively robustness since it can inject large-size noise into the local information necessary for reconstruction, thanks to its inherent masking, but is upper bounded on the Cutout, which discards a part of the image corresponding to the mask. Finally, the larger the training dataset size, the better the decoder model is trained, which reduces the overall reconstruction loss.