Privacy Leakage in Large Foundation Models

Dixi Yao

University of Chicago

- Research Direction: Privacy-preserving Machine Learning (over Distributed Systems)
- Large Foundation Models: LLM and Diffusion Model
- Evidence shows that they can memorize data well
- This raise concerns in privacy such as data leakage

- Potential Direction:
  - Resolve from Data
  - Resolve from System (Architecture)

Think from Data

Towards Privacy-Preserving Split Learning with ControlNet and Stable Diffusion

Accepted by WACV 2025

Motivation: Defence against inversion attacks when we train a ControlNet in Split Learning

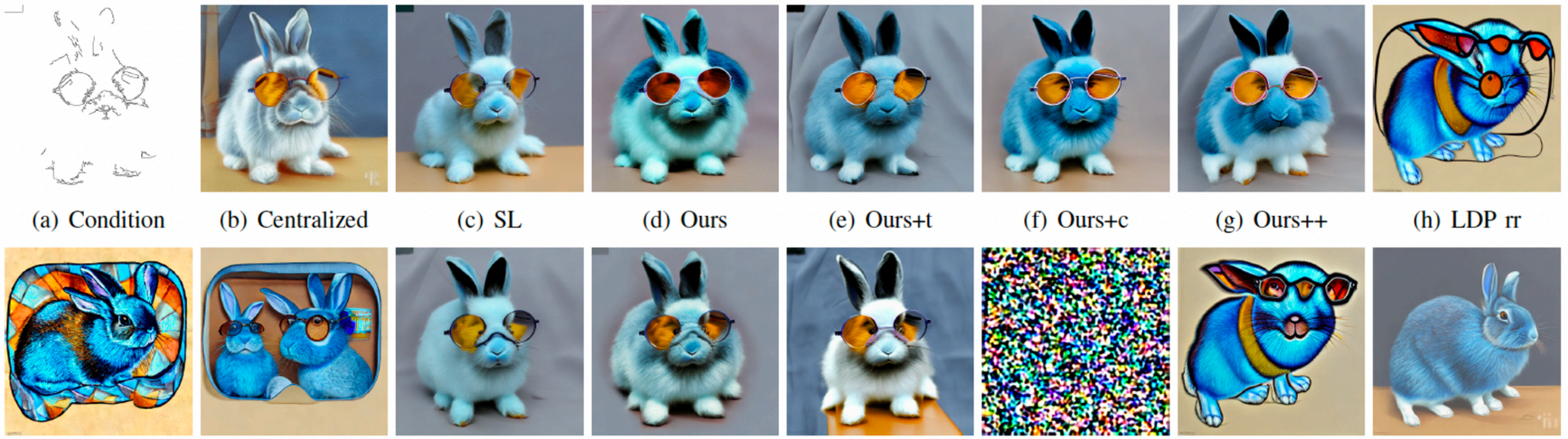
Research Question: How can we train ControlNet models while keeping users' data privacy, especially if such data is distributed over multiple client devices?

· Core Idea

The process of adding noise during the diffusion process can be treated as a process of adding differential privacy noise.

The idea is to construct a relationship between privacy budget and the scheduling of adding noise during diffusion processes.

· Results



(a) Condition (b) Centralized (c) SL (d) Ours (e) Ours+t (f) Ours+c (g) Ours++ (h) LDP rr

(i) LDP 0.1 (j) LDP 0.3 (k) LDP 0.5 (l) Add 1 (m) Add 50 (n) Mixup (o) PS (p) FedAvg

Fig. 11. **Image generation:** Images of higher quality means better. Randomly selected and non-cherry-picked examples of images generated with the Canny condition under different methods. The text prompt is *a blue rabbit with glasses*.

(a) k, $\epsilon_s = 2$ (b) k, $\epsilon_s = 0.3$ (c) β_0, $\epsilon_s = 2$ (d) β_0, $\epsilon_s = 0.3$

Fig. 16. Randomly selected and non-cherry-picked examples of generated images varying privacy budget with different k and β_0 in our method (Ours++). k and β_0 indicate that the privacy budgets change by altering only k or β_0 respectively compared to the default settings.

How Can We Extract the Private Data used for Fine-tuning a Large Language Model?

Assumption:

- A company or a user fine-tune a pre-trained large language model using private data.
- The private data has no overlap with the data used for pre-training.
- The fine-tuned model is also publicly released.
- The attackers do not have any knowledge of private data (data types, prefaces, suffices, etc.)

Experiment Design:

- Llama 3-8B Model; Fine-tuning following LoRA
- Dataset: Texts of the papers from ArXiv in 2024

Experiments

Experiment Name	Number of Samples Found in the Whole Dataset	Number of Samples Found in Finetuning Dataset	Number of Samples Found in Finetuning Dataset with zlib entropy >=0.51	Number of Samples Found in Finetuning Dataset without Duplicates
extraction	3 / 2000	3 / 2000	3 / 2000	1 / 2000
inversion_extraction	3 / 2000	3 / 2000	3 / 2000	1 / 2000
scalable_extraction	3 / 2000	3 / 2000	3 / 2000	1 / 2000
ours_extraction	484 / 2000	484 / 2000	414 / 2000	5 / 2000
ProPILE_BlackBox	0 / 100	0 / 100	0 / 100	0 / 100

- Extraction / Scalable Extraction: Construct prompts wit ‘startofthetex’ and ‘endofthetext’ and repeatedly let the model generate outputs. Assume these outputs are reconstructed private data.
- Inversion Extraction: Let the model randomly generate outputs and use an inversion network to infer the original inputs.
- Ours: Combine the first and the second method. First use constructed prompts to let model generate outputs and infer the inputs from these outputs using inversion networks.
- ProPILE: Use some templates and input them into the model and let them generate the outputs.

Takeaways

- We can extract private data used for fine-tuning large language model.
- Duplicating the samples in the training dataset can mitigate the issues:
  - We deliberately repeat some samples and these samples can be extracted easier (using fewer attempts of attacks)
  - We use another dataset, OpenRewriteEval, which has the overlap with the data used for pre-training. And more samples are extracted.
  - We can insert some non-private data and duplicate them.
- Shuffling the dataset cannot mitigate the issues. The possibility of a sample being leaked is irrelevant to the sequence it appears in the dataset. But some samples are much easier to be attacked
  - We repeated the experiments three times. And each time we shuffle the dataset: We found that the existence of leaked samples distributed evenly

First experiment ours_extraction
- Indices in the original dataset: {0, 1, 2, 3, 4}
- Indices in the finetuning sequence: {8, 23, 80, 139, 243}

Second experiment ours_extraction
- Indices in the original dataset: {0, 1, 2, 3, 4}
- Indices in the finetuning sequence: {46, 87, 137, 159, 172}

First experiment ours_extraction
- Indices in the original dataset: {0, 1, 2, 3, 4}
- Indices in the finetuning sequence: {34, 49, 95, 112, 185}

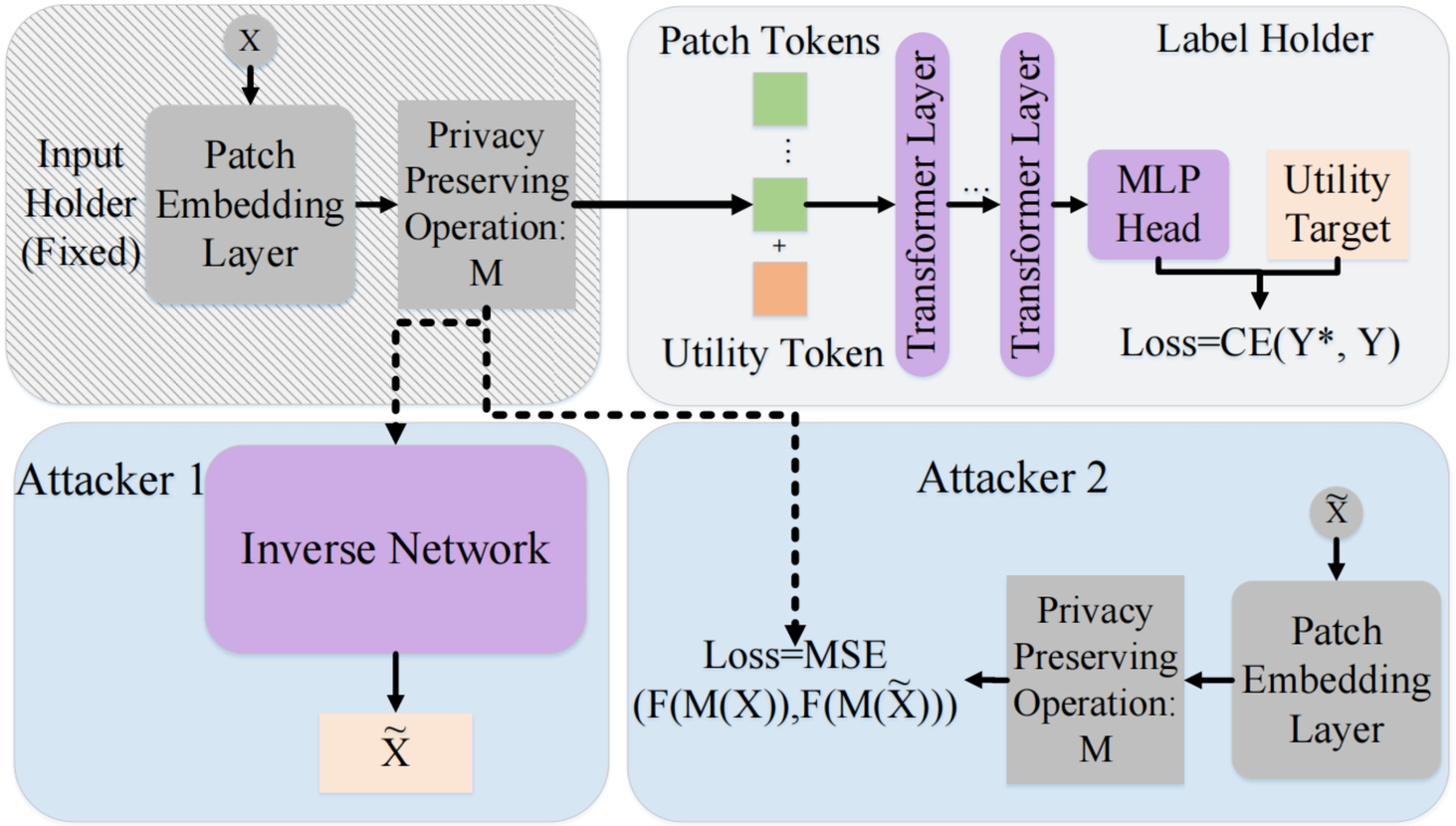
Think from System

Privacy-Preserving Split Learning via Patch Shuffling over Transformers

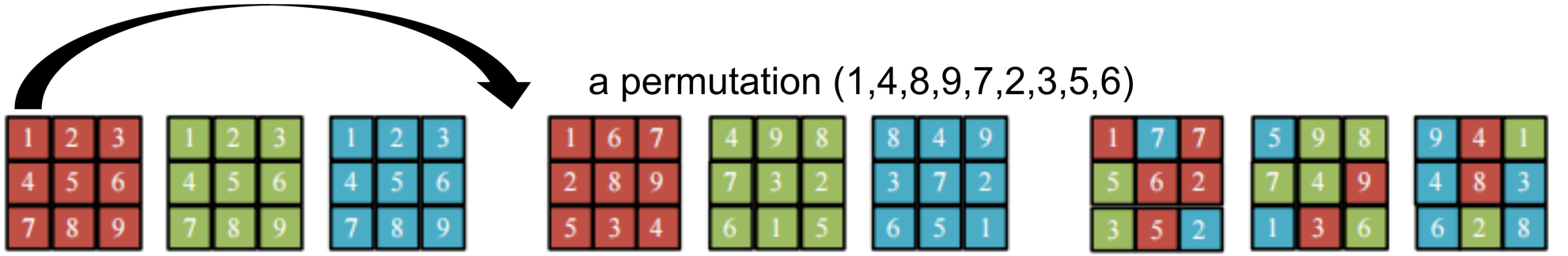
Published in 2022 IEEE ICDM, 2024 CVPR

Motivation: Defence against inversion attacks over ViT and images in split learning

Research Question: Whether we can leverage a special property of transformer models to keep users' data privacy in split learning?



· Core Method



· Results

➢Visualization effect of CelebA reconstruction

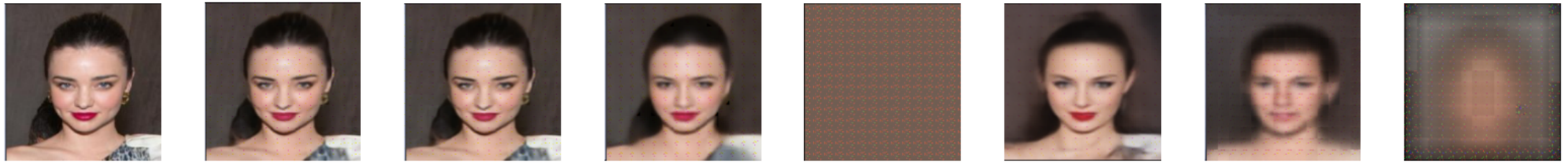


Fig. 6. Examples of reconstructed images by the black-box attack against different defence methods on CelebA.

Accuracy(%)	91.05	90.36	89.58	80.67	87.35	89.18	88.21

Is Split Learning Privacy-Preserving for Fine-tuning Large Language models

IEEE TBD, special issue for pre-trained large language models

Motivation: Split learning is vulnerable to a series of attacks for training conventional models such as 1-D CNNs and 2-D CNNs.

We focus mainly on the UnSplit Learning and attacking methods having the similar methodologies.

Reasons: Applicable in real-world scenarios in the context of text generation with large language models

Attackers have no pre-knowledge of private dataset; LLMs are pre-trained and publicly available.

TABLE 2: The effectiveness of improved UnSplit attack in the first and second iteration during split learning, on WikiText-103. Number of parameters are counted in million. Acc, R1 F1, R2 F1, RL F1, and RL sum F1 represent Accuracy, ROUGE-1 F1, ROUGE-2 F1, ROUGE-L F1, and ROUGE-L sum F1 respectively.

		First iteration					Second iteration				
	# Param	Acc%	R1 F1%	R2 F1%	RL F1%	RL sum F1%	Acc%	R1 F1%	R2 F1%	RL F1%	RL sum F1%
GPT-2	46.5	82.57	86.38	61.96	78.90	86.18	0.415	0.941	0.761	0.602	0.836
OPT	157.6	43.55	41.54	16.61	38.49	40.35	2.95	4.99	3.02	3.60	4.88
OpenChat 3.5	349.2	29.13	28.82	8.73	26.69	28.14	0.0	0.09	0.11	0.077	0.121
Qwen 1.5	361.8	11.72	16.92	9.04	13.54	16.92	1.17	0.17	0.0	0.14	0.17
Llama 2	333.6	36.08	45.79	31.98	45.79	45.73	0.879	1.01	0.0	0.864	1.01
Llama 3	743.6	19.82	25.57	10.46	23.23	25.25	0.12	0.348	0.0	0.349	0.349