You Can Use But Cannot Recognize: Preserving Visual Privacy in Deep Neural Networks

Qiushi Li, Yan Zhang, Ju Ren, Qi Li, Yaoxue Zhang

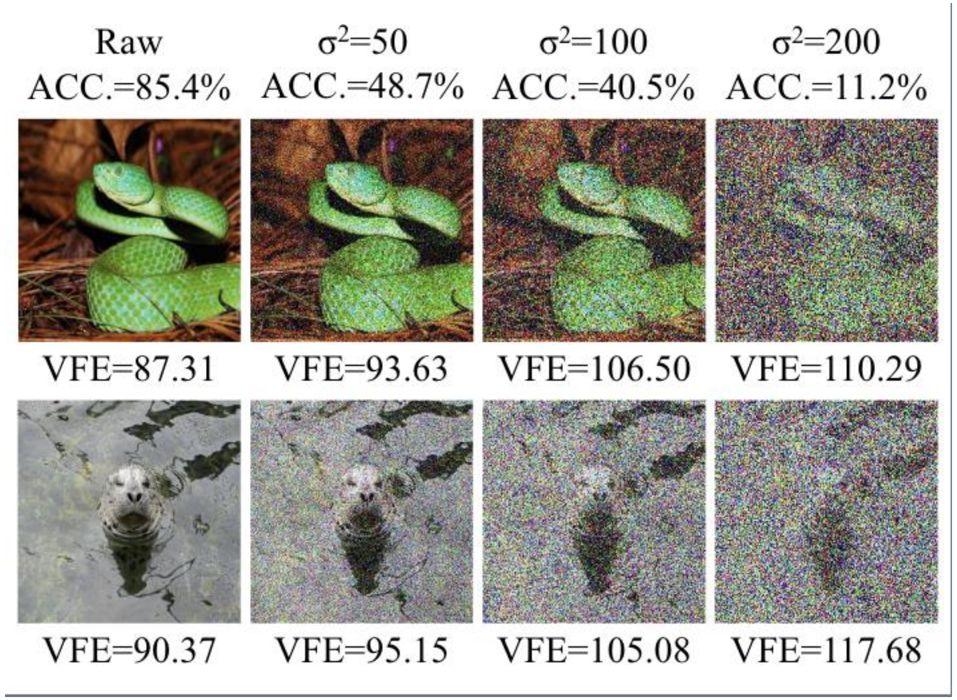
Tsinghua University

NDSS SYMPOSIUM/2024

Presented by Internet Society

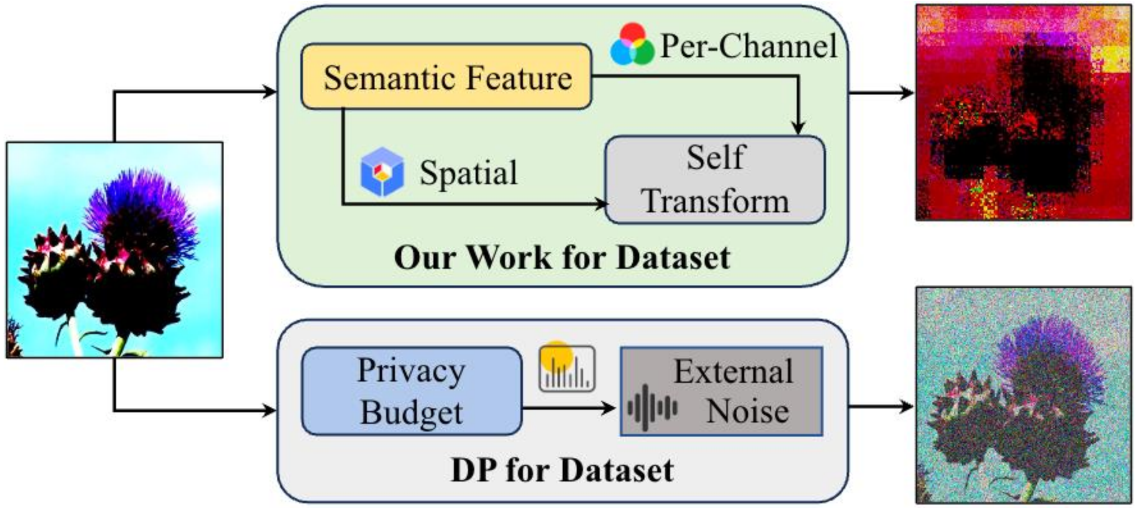
Problem:

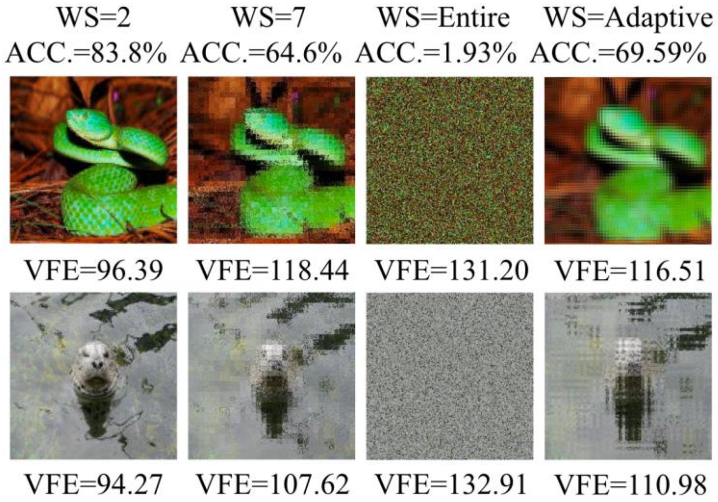
- Neural network models can leak the training datasets
- Existing privacy protection methods such as homomorphic encryption and differential privacy have their limitations.



Solution:

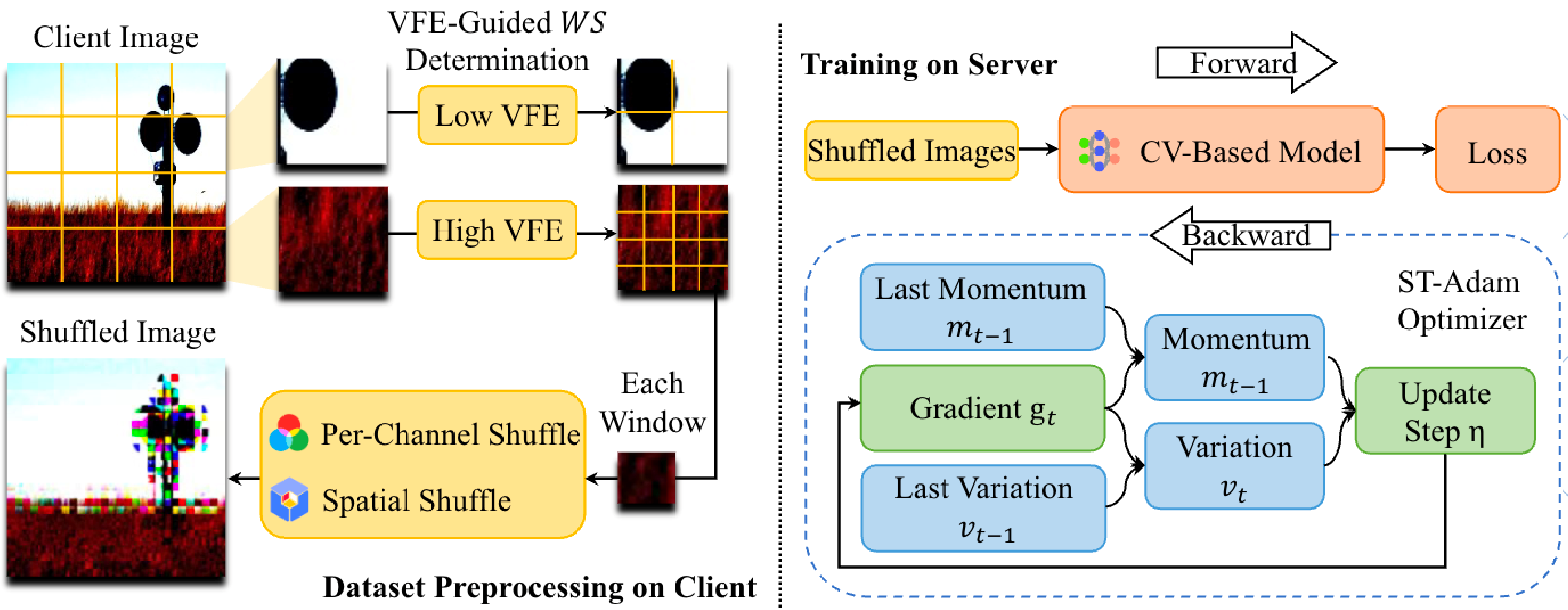
- Trade-off between privacy and loss of model performance
- Protect visual privacy of image data by shuffling





Architecture:

- **Using VFE to guide privacy-preserving image shuffle**
- **Improve the convergence speed of model training over the shuffled image data by ST-Adam Optimizer**



VFE:

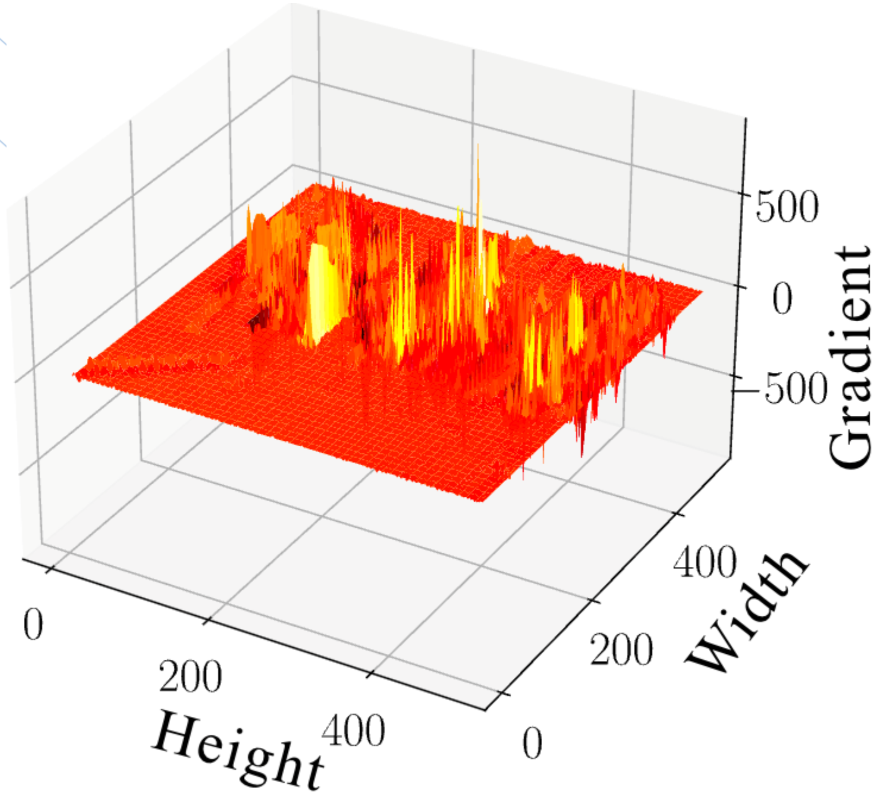
$$\nabla_x I(x,y) = I(x+1,y) - I(x,y), \quad x \in \{0,1,\ldots,N_1-1\}$$
$$\nabla_y I(x,y) = I(x,y+1) - I(x,y), \quad y \in \{0,1,\ldots,N_2-1\}$$

$$VFE_R(R_I) = \sum_{x=x_0}^{x_0+w-1} \sum_{y=y_0}^{y_0+h-1} \left(\nabla_x I(x,y)^2 + \nabla_y I(x,y)^2\right)$$

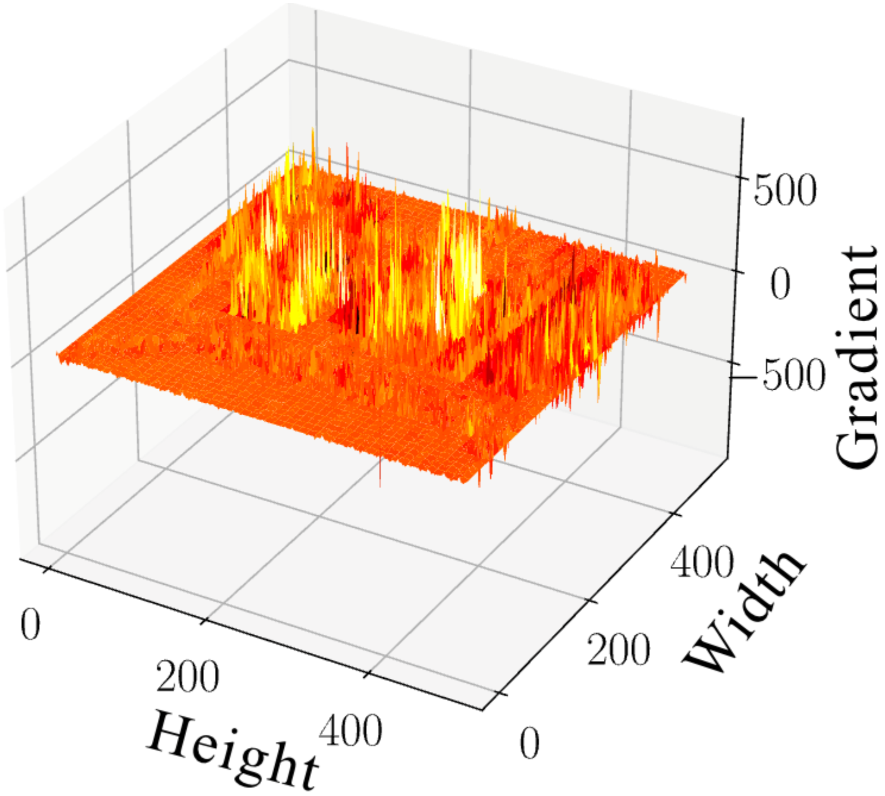
$$VFE(I) = \frac{F}{N_1 N_2} \sum^{R_I \in \boldsymbol{R_I}} VFE_R(R_I)$$

Challenge of training on the mixed dataset:

- models struggling to converge due to gradient oscillation



Original Gradient



VisualMixed Gradient

NDSS SYMPOSIUM/2024

Presented by Internet Society

#NDSSSymposium2024

ST-Adam Optimizer:

- **The update rules of ST-Adam Optimizer**

(1) calculates the gradient of the loss function

$$g_t = \nabla f(w_t)$$

(2) calculate the momentum by hyperparameter β

$$m_t = \beta \times m_{t-1} + (1 - \beta) \times g_t$$

(3) calculate the daptive learning rate by hyperparameter γ

$$v_t = \gamma \times v_{t-1} + (1 - \gamma) \times g_t^2$$

(4) update the parameters of models

$$w_{t+1} = w_t - \eta * \frac{m_t}{\sqrt{(v_t)} + \epsilon}$$

ST-Adam Optimizer:

- **Why ST-Adam Optimizer?**

(1) First define

$$\Delta w_t = w_t - w^*, \quad \Delta f_t = f(w_t) - f(w^*)$$

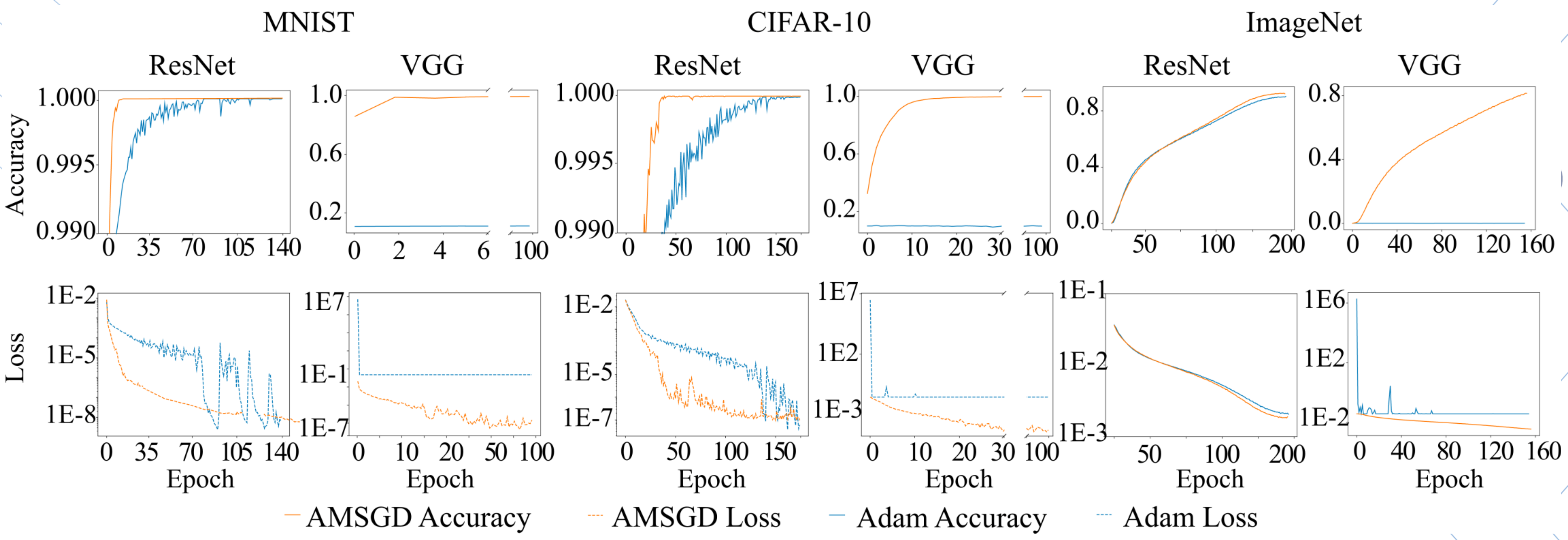
(2) According to Jensen's inequality

$$\Delta f_t \leq g_t^T \times \Delta w_t$$

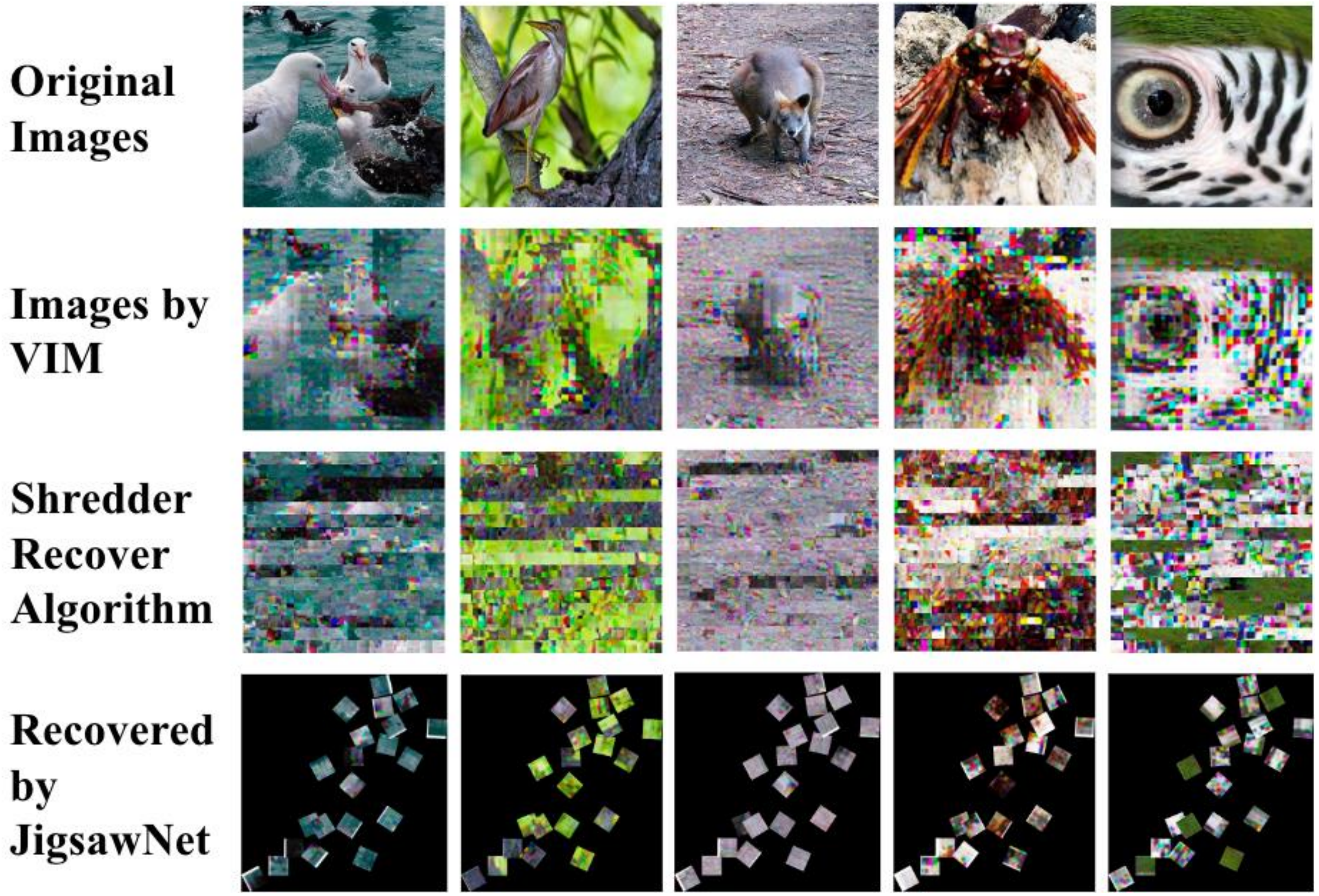
(3) Substituting the update rule of ST-Adam

$$\Delta f_t \leq \left(\frac{m_t}{\sqrt{v_t} + \epsilon}\right)^T \times \Delta w_t$$

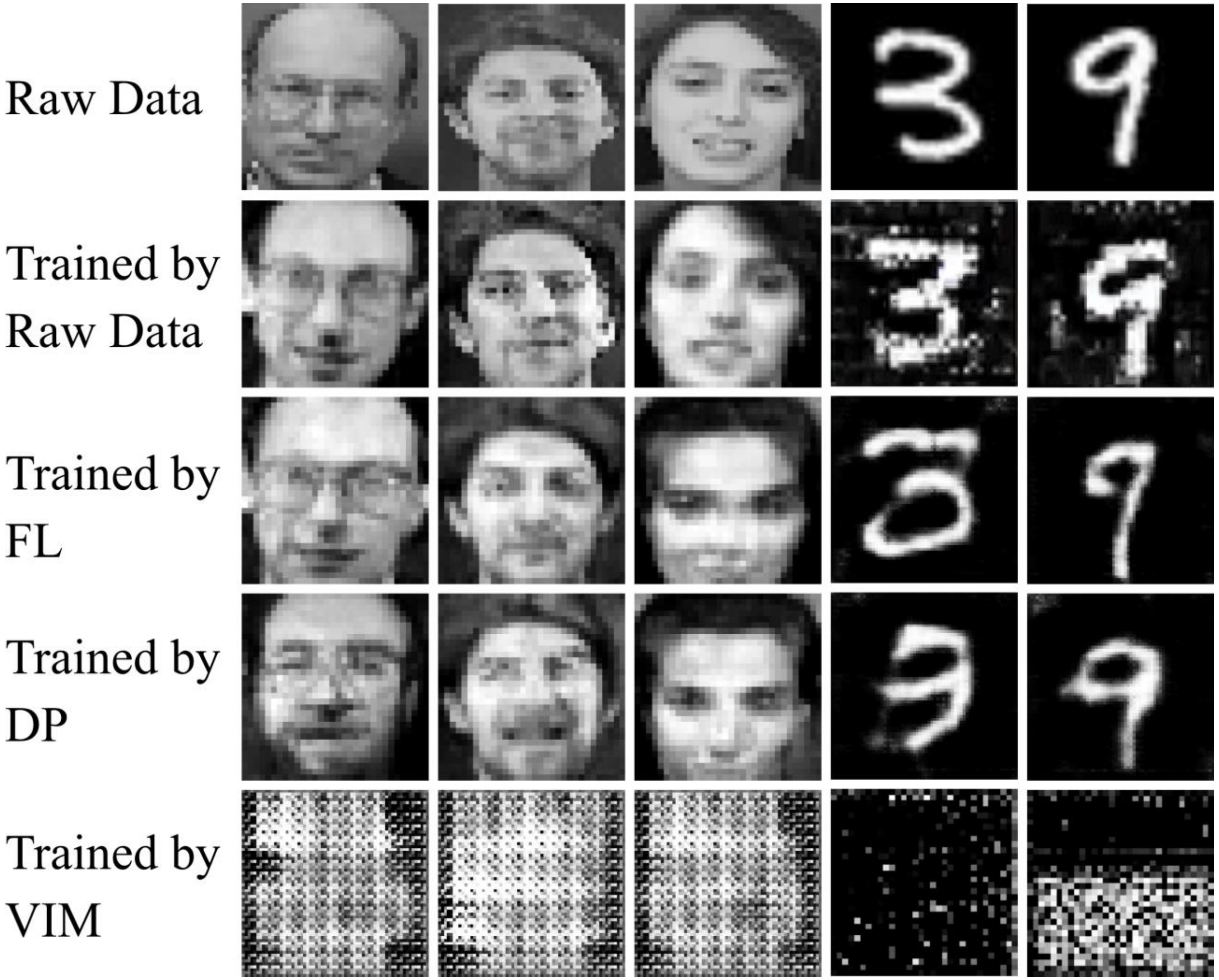
Validation on ST-Adam Optimizer:



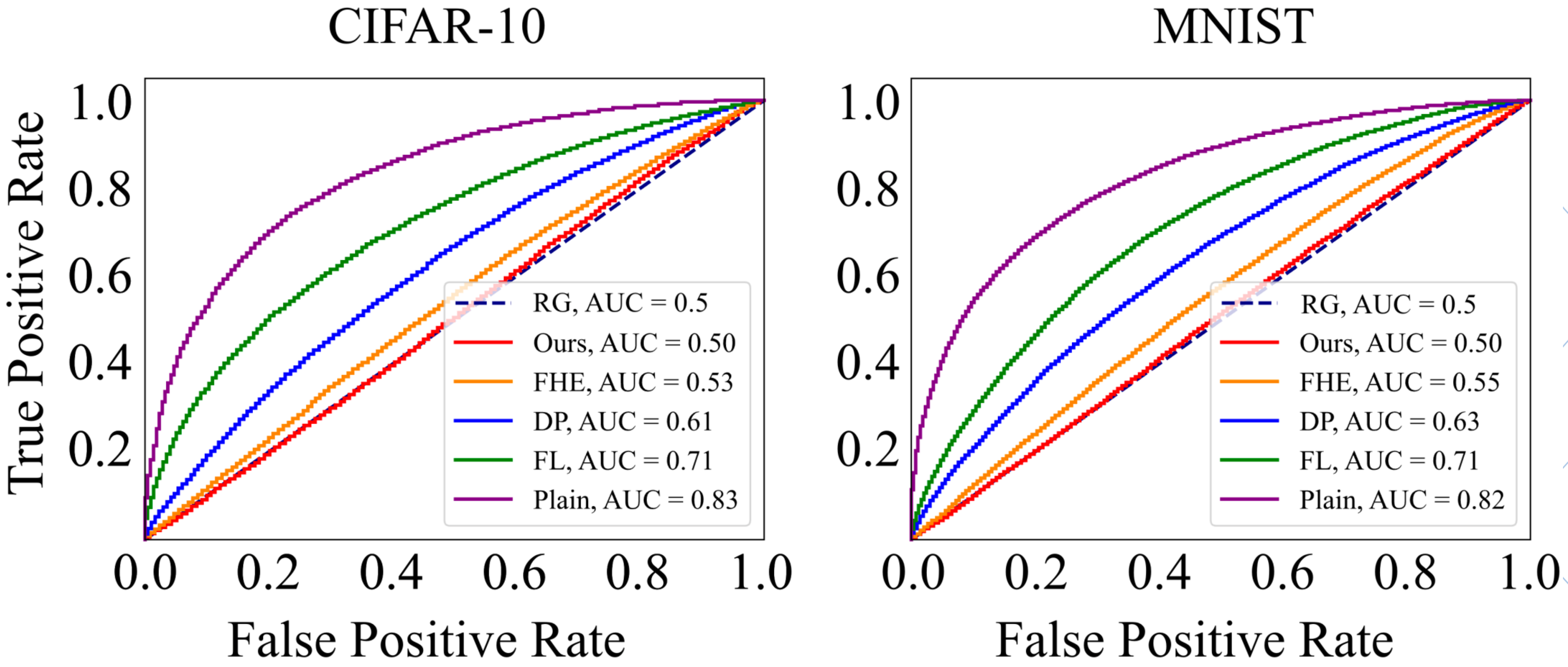
Defend Against Heuristic Attacks:



Defend Against GAN:



Defend Against Membership Inference:



Performance:

TABLE II: Accuracy of trained models with different datasets.

Model	MNIST					CIFAR-10					ImageNet-100[5]			
Method	Plain	VIM	DP	FHE[4]	InstaHide [24]	Plain	VIM	DP	FHE[4]	InstaHide	Plain	VIM	DP	InstaHide
Privacy	✗	✓	◯[1]	✓	◯[7]	✗	✓	◯[1]	✓	◯[7]	✗	✓	◯[1]	◯[7]
ViT-B [9]	99.87%	99.14%	-[2]	-[4]	9.97%	98.63%	92.35%	-[2]	-[4]	10.03%	74.54%	72.98%	-[2]	1.03%
Swin-T [35]	98.72%	98.70%	-[2]	-[4]	10.16%	92.33%	85.73%	-[2]	-[4]	9.82%	84.80%	81.12%	-[2]	0.10%
ResNet [18]	99.27%	98.81%	61.36%[3]	-[4]	98.79%	97.23%	90.15%	62.74%[3]	87.84%[6]	90.04%	90.34%	83.78%	60.82%[3]	31.08%
ShuffleNet [52]	98.93%	97.19%	58.91%[3]	-[4]	96.27%	86.87%	84.07%	52.06%[3]	-[4]	84.97%	85.34%	83.64%	48.75%[3]	29.78%
MobileNet [22]	97.21%	97.20%	51.48%[3]	-[4]	97.13%	81.37%	81.02%	59.77%[3]	-[4]	75.53%	82.94%	81.38%	48.57%[3]	30.94%
VGG [44]	99.51%	98.12%	69.34%[3]	[4]	98.05%	82.64%	82.63%	53.89%[3]	84.76%[6]	82.57%	74.02%	73.88%	43.56%[3]	1.38%

TABLE III: Throughput (images per second) of different methods on different datasets.

Method	Privacy	ShuffleNet [52]	VGG [44]	ResNet [18]	MobileNet [22]	ViT-B [9]	Swin-T [35]
Plain	✗	1088.9	404.7	600.2	1070.8	322.2	472.3
DP [10]	◯[1]	212.3 [-80.5%]	66.1 [-83.7%]	187.8 [-68.7%]	92.1 [-91.4%]	-[2]	-[2]
FL[3][38]	◯[1]	291.8 [-73.2%]	385.2 [4.8%]	565.1 [5.8%]	1008.7 [-5.8%]	-[4]	-[4]
DP + FL[3] [10, 38]	◯[1]	10.9 [-99.0%]	2.0 [-99.5%]	7.4 [-98.8%]	12.8 [-98.8%]	-[4]	-[4]
FHE[5][39]	✓	0.006[-99.9%]	0.0009 [-99.9%]	0.0005[-99.9%]	0.005 [-99.9%]	0.000074 [-99.9%]	0.00045 [-99.9%]
InstaHide [24]	◯[6]	1087.1 [-0.17%]	399.2 [-1.4%]	594.1 [-1.0%]	1062.3 [-7.9%]	315.8 [-2.0%]	458.3 [-3.0%]
VIM	✓	1080.3 [-0.8%]	401.9 [-0.6%]	595.4 [-0.8%]	1062.1 [-0.8%]	319.9 [-0.7%]	466.1 [-1.3%]

Performance:

TABLE IV: Federated learning accuracy of ResNet50 on ImageNet via plain scheme and VIM scheme.

Model	Method	MNIST	CIFAR10
ResNet [18]	FL FL+VIM	99.28% 94.39%	70.83% 66.11%
VGG [44]	FL FL+VIM	99.48% 97.25%	77.29% 76.87%
MobileNet [22]	FL FL+VIM	99.23% 97.44%	75.26% 73.11%
ShuffleNet [52]	FL FL+VIM	99.20% 97.33%	72.63% 72.08%
Swin-T [35]	FL FL+VIM	95.47% 93.71%	67.53% 61.37%
ViT-B [9]	FL FL+VIM	92.29% 87.23%	60.03% 57.70%

TABLE V: Knowledge distillation [19] accuracy of ResNet50 on ImageNet-100 via different training schemes.

Model	Top-1	Top-3	Top-5	Top-10
TO-Resnet50 **TV**[1]Resnet50	74.55% 66.45%	88.42% 82.67%	92.02% 87.24%	95.23% 91.78%
SO[2]MobileNetv3 **SV**-MobileNetv3	21.1% 18.9%	44.0% 43.1%	53.6% 53.4%	62.4% 62.4%

TABLE VI: Experimental results on VOC dataset.

Model	**Method**	**Precision**	**Recall**	**mAP@50**
YOLO v5 [26]	Plain VIM	0.601 0.602	0.534 0.415	0.562 0.441
SSD [34]	Plain VIM	0.631 0.556	0.594 0.418	0.504 0.372
EfficientDet [46]	Plain VIM	0.817 0.735	0.660 0.419	0.765 0.505

Performance:

TABLE IV: Federated learning accuracy of ResNet50 on ImageNet via plain scheme and VIM scheme.

Model	Method	MNIST	CIFAR10
ResNet [18]	FL FL+VIM	99.28% 94.39%	70.83% 66.11%
VGG [44]	FL FL+VIM	99.48% 97.25%	77.29% 76.87%
MobileNet [22]	FL FL+VIM	99.23% 97.44%	75.26% 73.11%
ShuffleNet [52]	FL FL+VIM	99.20% 97.33%	72.63% 72.08%
Swin-T [35]	FL FL+VIM	95.47% 93.71%	67.53% 61.37%
ViT-B [9]	FL FL+VIM	92.29% 87.23%	60.03% 57.70%

TABLE V: Knowledge distillation [19] accuracy of ResNet50 on ImageNet-100 via different training schemes.

Model	Top-1	Top-3	Top-5	Top-10
TO-Resnet50 **TV**[1]Resnet50	74.55% 66.45%	88.42% 82.67%	92.02% 87.24%	95.23% 91.78%
SO[2]MobileNetv3 **SV**-MobileNetv3	21.1% 18.9%	44.0% 43.1%	53.6% 53.4%	62.4% 62.4%

TABLE VI: Experimental results on VOC dataset.

Model	**Method**	**Precision**	**Recall**	**mAP@50**
YOLO v5 [26]	Plain VIM	0.601 0.602	0.534 0.415	0.562 0.441
SSD [34]	Plain VIM	0.631 0.556	0.594 0.418	0.504 0.372
EfficientDet [46]	Plain VIM	0.817 0.735	0.660 0.419	0.765 0.505