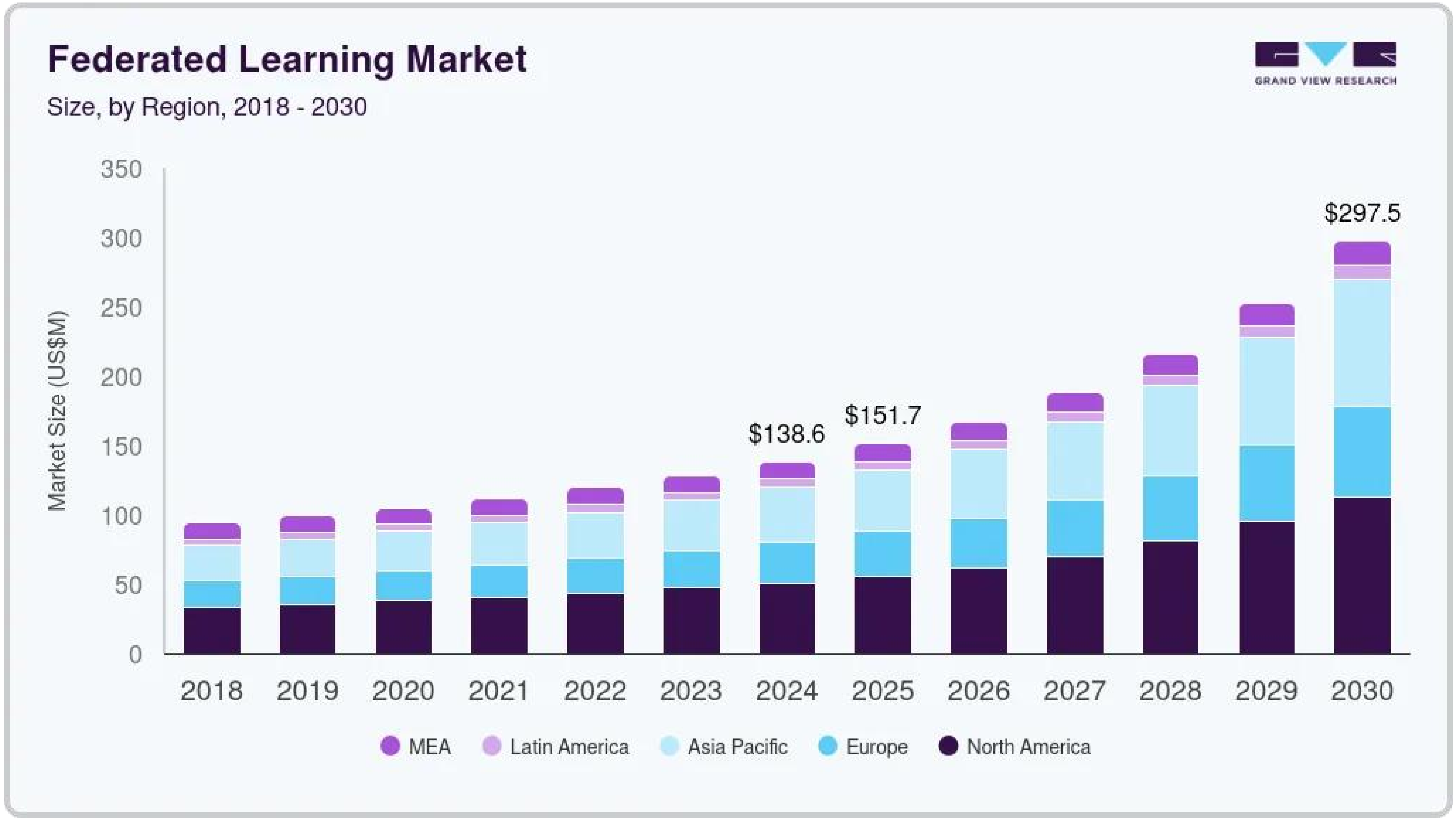
SVDefense: Effective Defense against Gradient Inversion Attacks via Singular Value Decomposition

Chenxiang Luo[1], David Yau[2], **Qun Song**[1]

[1]City University of Hong Kong

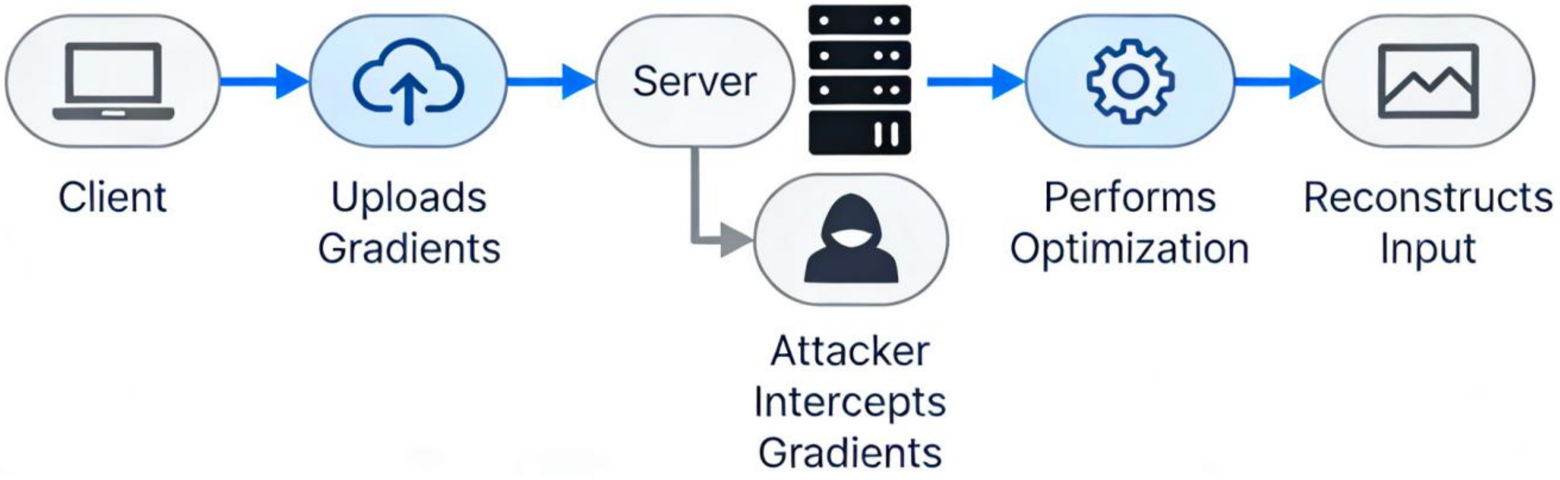
[2]Singapore University of Technology and Design

Federated Learning Market is Growing



Source: https://www.grandviewresearch.com/industry-analysis/federated-learning-market-report

Gradient Inversion Attacks (GIAs)



Optimization Objective

Reconstruct private input data (x) from shared gradients (∇L) by solving an optimization problem:

Minimize: $D(\nabla L(x), \nabla L(x`)) + \lambda \cdot R(x`)$

Existing Defenses against GIAs

- Encryption-based [CCS’17, USENIX’20, PMLR’22]
  - Employ cryptographic techniques to protect client updates.
  - Introduce significant overhead.

- Perturbation-based
  - *Input perturbation* modifies the local training data [PMLR’20, CVPR’21, AAAI’24].
  - *Gradient perturbation* modifies the local gradients [ICDCS’21, INFOCOM’23, NDSS’25].
  - *Training perturbation* perturbs local training processes [WACV’22, AAAI’23, ICCV’23].
  - Struggle to balance good defense performance and model utility.

- Pruning-based [NeurIPS ’19, CVPR’21,AAAI’24]
  - Selectively remove gradient components.

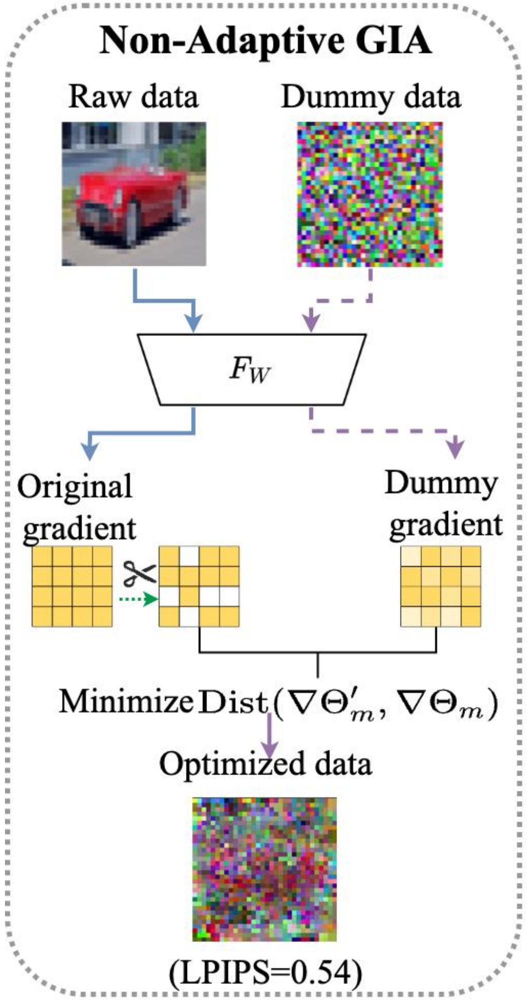
- Compression-based [ESORICS’23, CVPR’23]
  - Mitigate information leakage by compressing gradients.

Existing Defenses are Vulnerable to Adaptive Attack

- Many defenses can be bypassed by adaptive adversaries who have obtained the defense details.

Existing Defenses are Vulnerable to Adaptive Attack

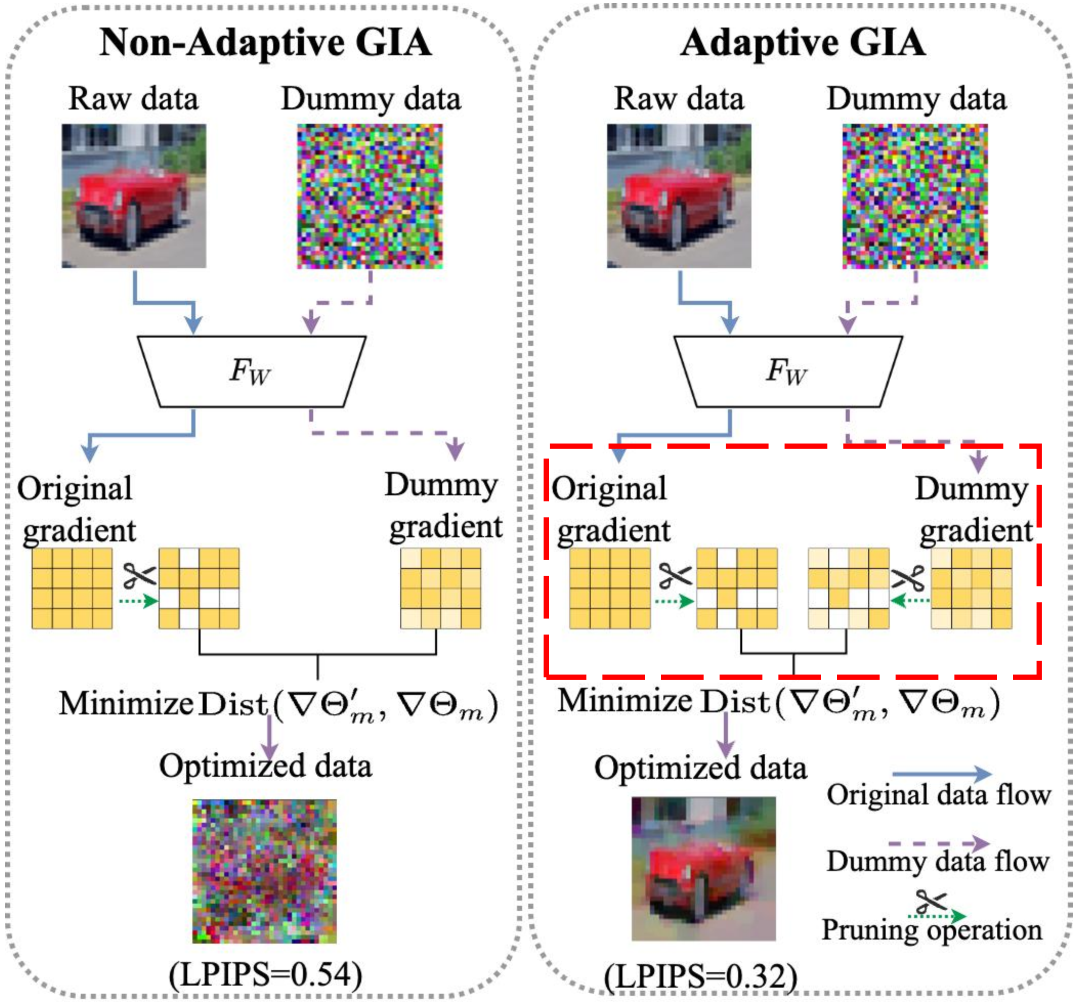
- Take the pruning-based defense as an example.



Existing Defenses are Vulnerable to Adaptive Attack

- Take the pruning-based defense as an example.

- Adaptive attacker could apply the same defense operations when attacking and get better reconstruction performance.



Motivation Study

- **Pruning-based Defenses**:
  - Detect zero values in gradients; apply the same pruning to dummy gradients.
  - Prune [NeurIPS’19], Soteria [CVPR’21], and DGP [AAAI’24]

- **Random Variable-based Defenses**:
  - Initialize a dummy random vector, optimize it with dummy inputs during attack construction.
  - PRECODE [WACV’22]

- **CENSOR** [NDSS’25]:
  - Perturb gradients for few initial epochs
  - Attack in undefended epochs.

Motivation: Results

Defense Performance of Different Methods Under Non-adaptive and Adaptive GIAs

Defense Methods	Non-adaptive		Adaptive	
	PSNR	LPIPS	PSNR	LPIPS
CENSOR	8.1940	0.6958	16.4071	0.2881
PRECODE	3.5659	0.7668	57.4165	0.0001
Prune	12.9257	0.4993	36.1273	0.0221
Soteria	10.8145	0.6481	38.7447	0.0161
DGP	9.7334	0.6187	35.6383	0.0254

*: Higher PSNR and Lower LPIPS Values Mean Stronger Attack Performance

Key Insignts

Pruning-Based Defenses

Partial Protection

Adaptive attackers exploit unaffected gradient components to reconstruct inputs effectively.

Random Variable Defenses

Reversible

Random variables used in protection can be obtained via variable recovery

Episodic Defenses

Short-Term Protection

Methods like CENSOR leave later epochs exposed, while continuous defense application degrades model utility, making it difficult to strike a balance.

Motivation: Introducing Truncated SVD

1) Irreversibly affects all gradients, 2) prudently truncates the gradients while preserving model utility

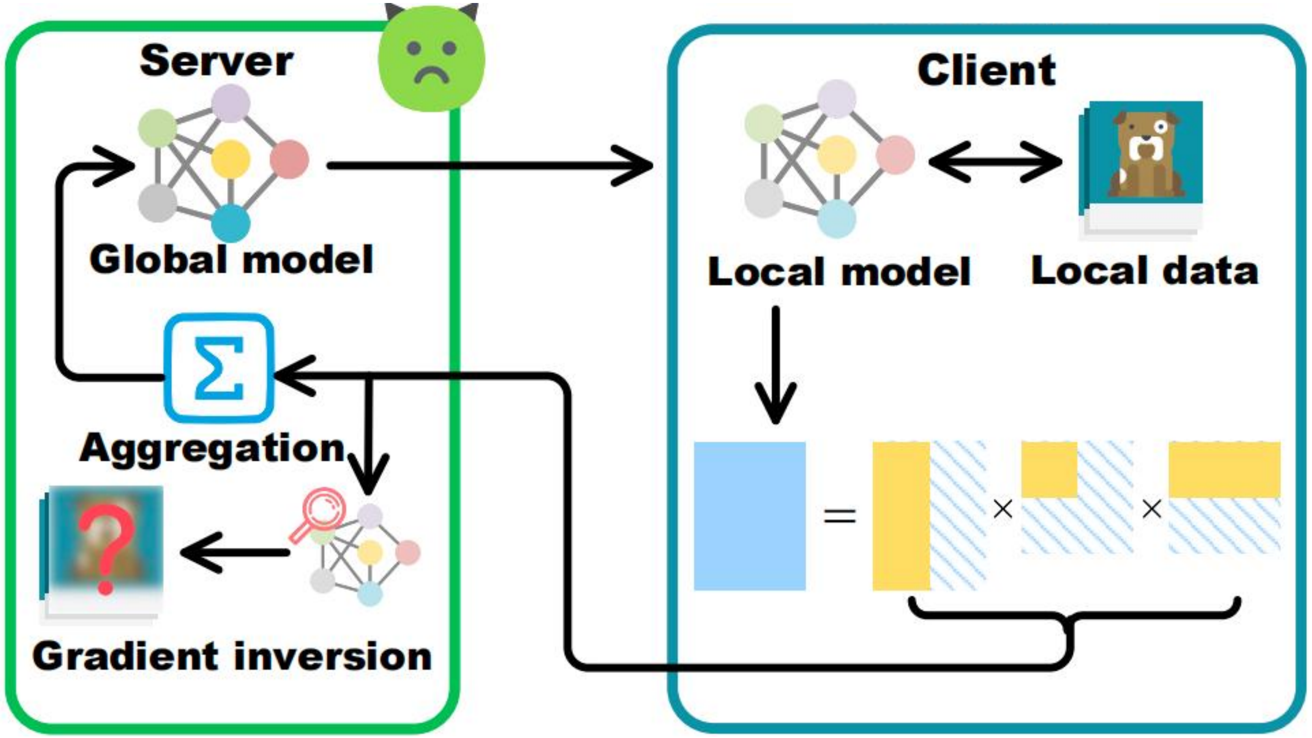
Singular Value Decomposition (SVD)

- Task: Approximate a matrix W with a lower-rank matrix W' to minimize the difference, with $\mathrm{rank}(W') \leq k$.

- Decomposition: $U\Sigma V^T$ = SVD(W)
  - U: Left singular vectors.
  - Σ: Singular values.
  - V^T: Right singular vectors transpose.

- Truncated SVD: Keep top k singular values and vectors → $W' = U' \Sigma' V'^T$. Choose k by energy threshold T.

Overview of SVDefense



Impact of Non-IID Data on GIAs

- Clients with higher degrees of class imbalance are more vulnerable to attacks.
- Inadvisable to treat clients with varying degrees of class imbalance uniformly.

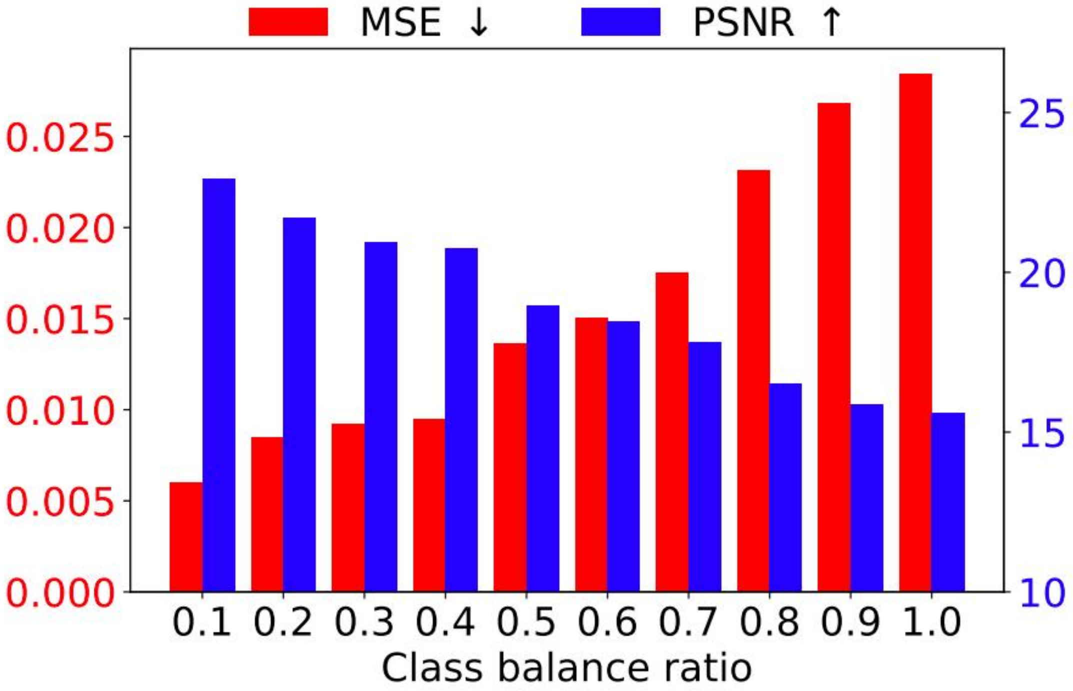


Figure 1: Impact of class imbalance on attack effectiveness.

*: Lower MSE and higher PSNR indicate stronger attack performance.

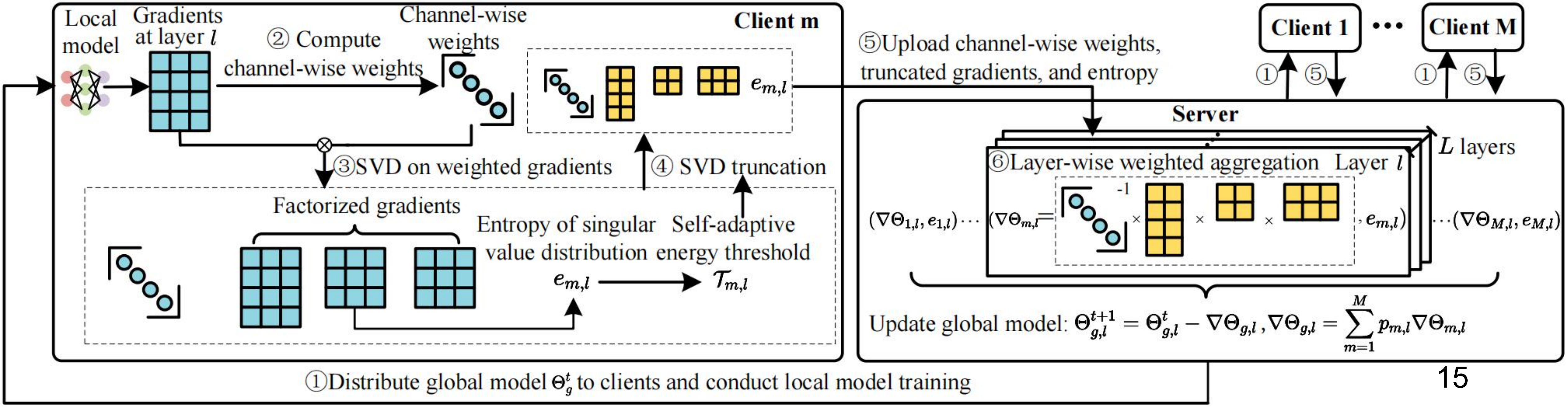
Key Challenges

C1: Adaptive Thresholding

C2: Privacy-Utility Trade-off

C3: Robust Aggregation

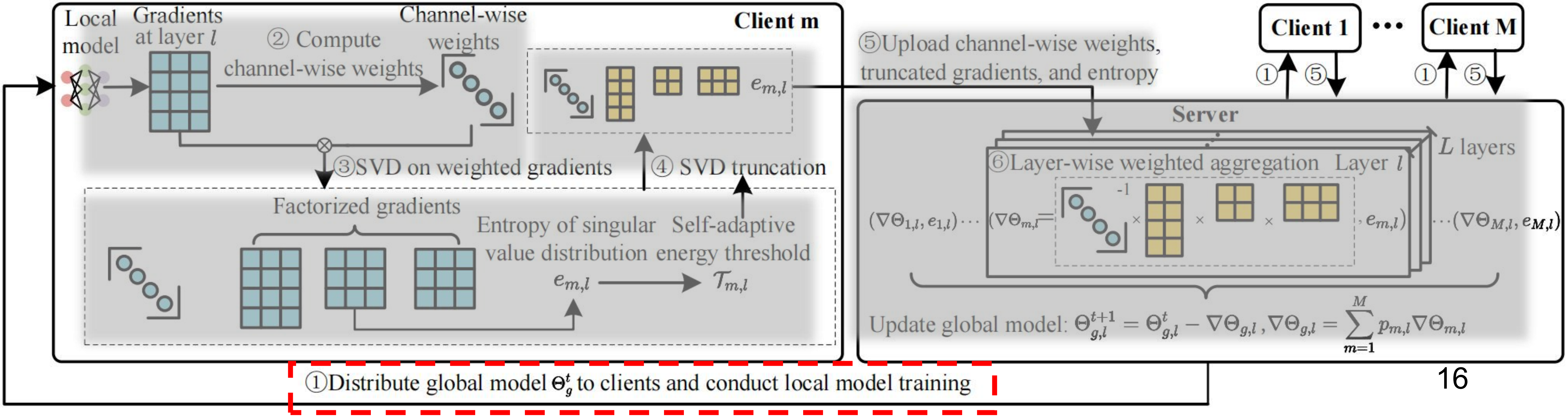
System Design: Overview


Local model
Gradients at layer l
② Compute channel-wise weights
Channel-wise weights
Client m
$e_{m,l}$
③SVD on weighted gradients
④ SVD truncation
Factorized gradients
Entropy of singular value distribution
Self-adaptive energy threshold
$e_{m,l}$
$\mathcal{T}_{m,l}$
⑤Upload channel-wise weights, truncated gradients, and entropy
Client 1
•••
Client M
①
⑤
①
⑤
Server
L layers
⑥Layer-wise weighted aggregation
Layer l
$(\nabla\Theta_{1,l}, e_{1,l})\cdots$
$(\nabla\Theta_{m,l}=$
-1
×
×
$, e_{m,l})$
$\cdots(\nabla\Theta_{M,l}, e_{M,l})$
Update global model: $\Theta_{g,l}^{t+1} = \Theta_{g,l}^{t} - \nabla\Theta_{g,l}, \nabla\Theta_{g,l} = \sum_{m=1}^{M} p_{m,l}\nabla\Theta_{m,l}$
①Distribute global model Θ_g^t to clients and conduct local model training

System Design: Overview

1. Local Training

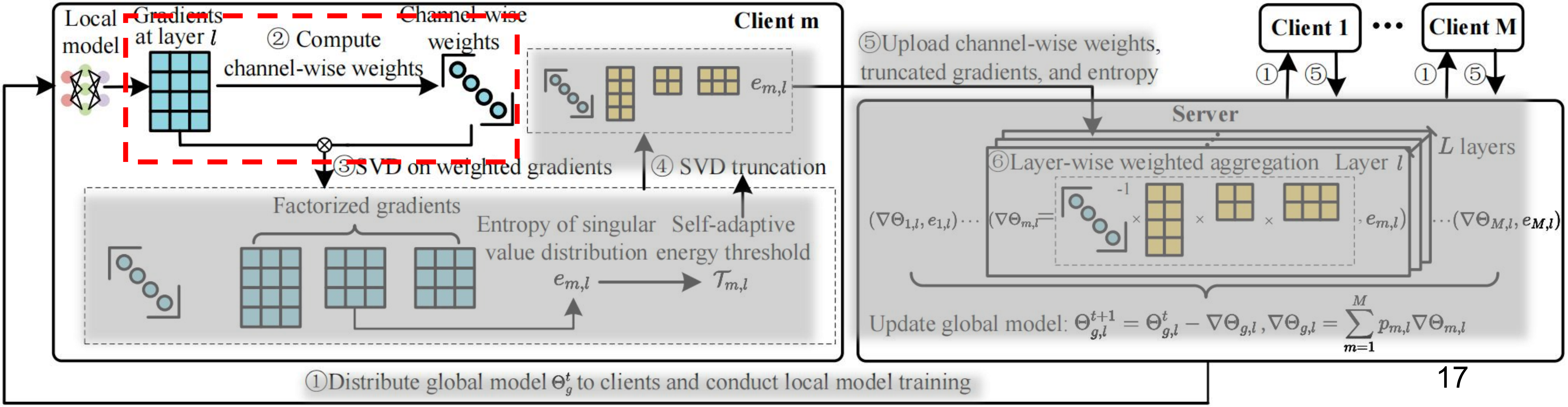
Clients train local models using their private data.


Local model
Gradients at layer l
② Compute channel-wise weights
Channel-wise weights
Client m
$e_{m,l}$
③SVD on weighted gradients
④ SVD truncation
Factorized gradients
Entropy of singular value distribution
Self-adaptive energy threshold
$e_{m,l}$
$\mathcal{T}_{m,l}$
⑤Upload channel-wise weights, truncated gradients, and entropy
Client 1
•••
Client M
①
⑤
①
⑤
Server
L layers
⑥Layer-wise weighted aggregation Layer l
$(\nabla\Theta_{1,l}, e_{1,l})\cdots$
$(\nabla\Theta_{m,l}=$
-1
×
×
$, e_{m,l})$
$\cdots(\nabla\Theta_{M,l}, e_{M,l})$
Update global model: $\Theta_{g,l}^{t+1}=\Theta_{g,l}^{t}-\nabla\Theta_{g,l}, \nabla\Theta_{g,l}=\sum_{m=1}^{M}p_{m,l}\nabla\Theta_{m,l}$
①Distribute global model Θ_g^t to clients and conduct local model training

System Design: Overview

2. Compute Channel-Wise Weights

Each client computes channel-wise weights based on gradient magnitudes.

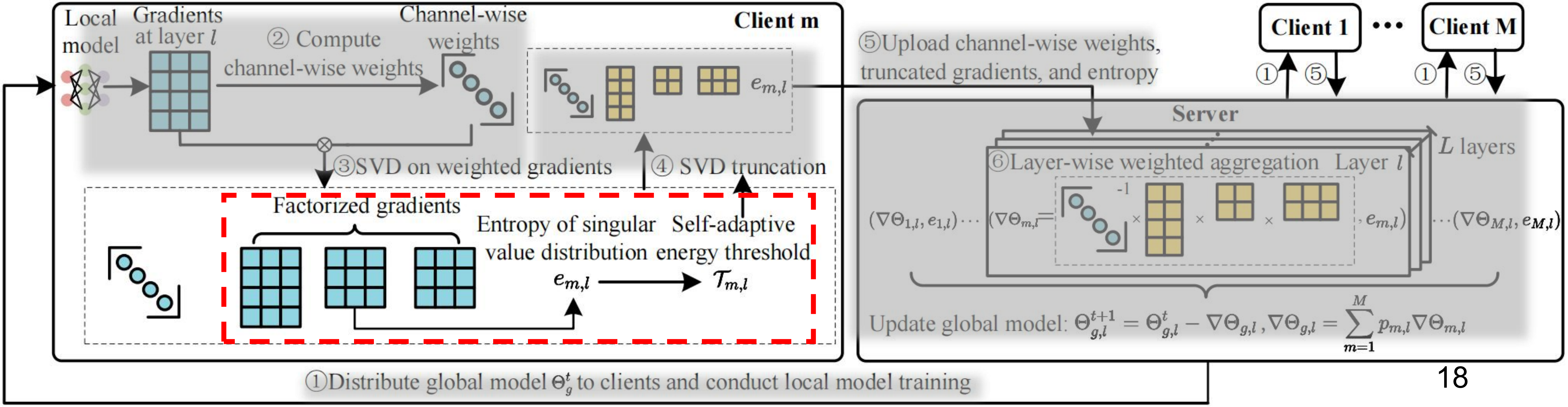


Local model
Gradients at layer l
② Compute channel-wise weights
Channel-wise weights
Client m
$e_{m,l}$
③SVD on weighted gradients
④ SVD truncation
Factorized gradients
Entropy of singular value distribution
Self-adaptive energy threshold
$e_{m,l}$
$\mathcal{T}_{m,l}$
⑤Upload channel-wise weights, truncated gradients, and entropy
Client 1
Client M
① ⑤
① ⑤
Server
L layers
⑥Layer-wise weighted aggregation
Layer l
$(\nabla\Theta_{1,l}, e_{1,l})\cdots$
$(\nabla\Theta_{m,l}=$
-1
×
×
$, e_{m,l})$
$\cdots(\nabla\Theta_{M,l}, e_{M,l})$
Update global model: $\Theta^{t+1}_{g,l} = \Theta^{t}_{g,l} - \nabla\Theta_{g,l}, \nabla\Theta_{g,l} = \sum_{m=1}^{M} p_{m,l}\nabla\Theta_{m,l}$
①Distribute global model Θ^t_g to clients and conduct local model training

System Design: Overview

3. SVD & Adaptive Threshold

Clients perform SVD on weighted gradients and calculate entropy-based adaptive energy threshold.

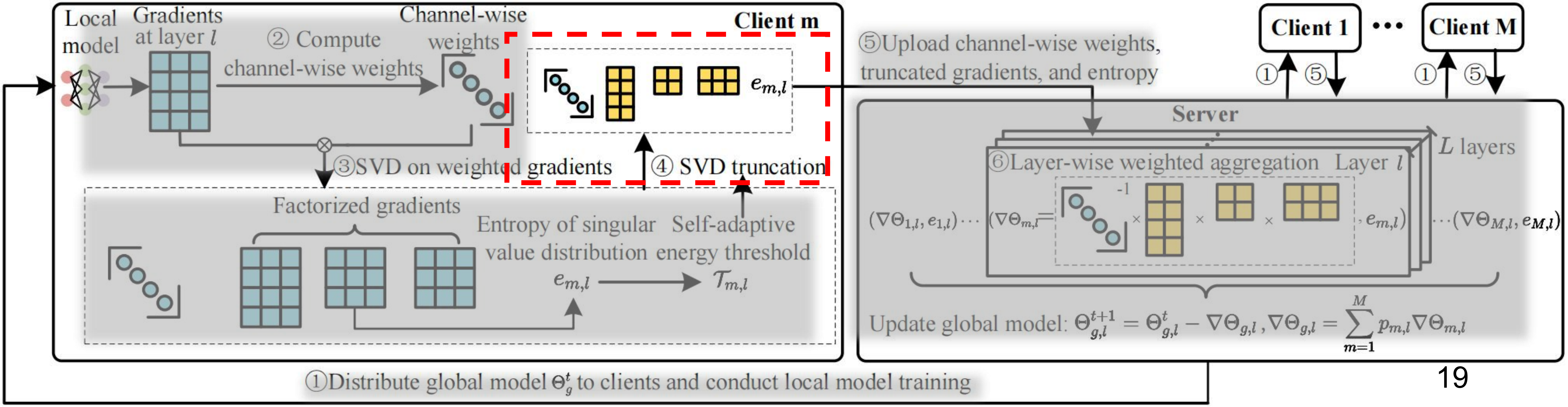


Local model
Gradients at layer l
② Compute channel-wise weights
Channel-wise weights
Client m
$e_{m,l}$
③SVD on weighted gradients
④ SVD truncation
Factorized gradients
Entropy of singular value distribution
Self-adaptive energy threshold
$e_{m,l}$
$\mathcal{T}_{m,l}$
⑤Upload channel-wise weights, truncated gradients, and entropy
Client 1
•••
Client M
①
⑤
①
⑤
Server
L layers
⑥Layer-wise weighted aggregation
Layer l
$(\nabla\Theta_{1,l}, e_{1,l})\cdots$
$(\nabla\Theta_{m,l} =$
-1
×
×
$, e_{m,l})$
$\cdots(\nabla\Theta_{M,l}, e_{M,l})$
Update global model: $\Theta_{g,l}^{t+1} = \Theta_{g,l}^{t} - \nabla\Theta_{g,l}, \nabla\Theta_{g,l} = \sum_{m=1}^{M} p_{m,l}\nabla\Theta_{m,l}$
①Distribute global model Θ_g^t to clients and conduct local model training

System Design: Overview

4. Gradient Truncation

Gradients are truncated according to the client-specific threshold.

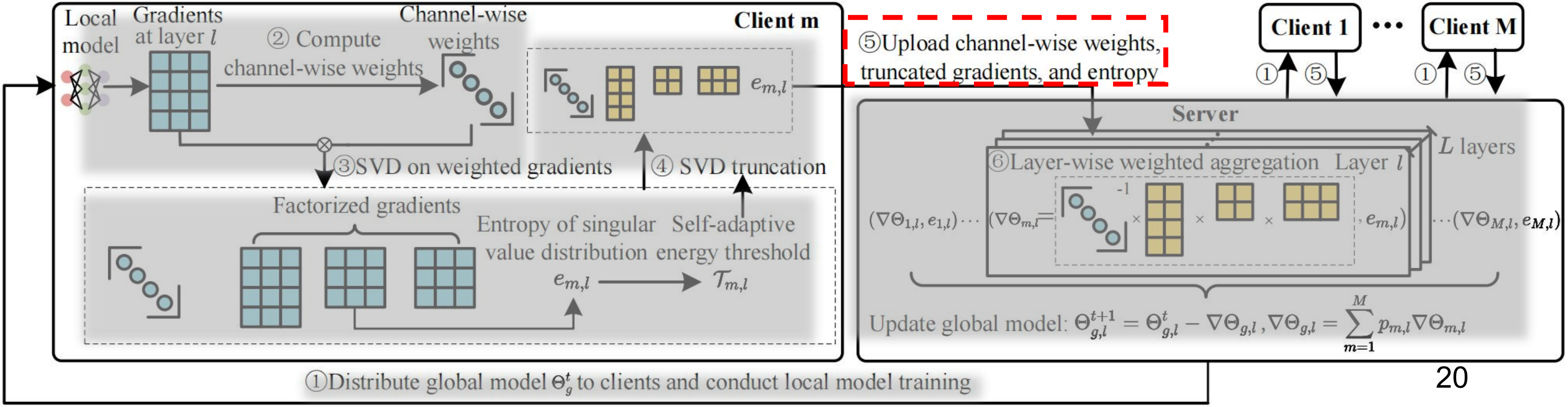


Local model
Gradients at layer l
② Compute channel-wise weights
Channel-wise weights
Client m
$e_{m,l}$
③SVD on weighted gradients
④ SVD truncation
Factorized gradients
Entropy of singular value distribution
Self-adaptive energy threshold
$e_{m,l}$
$\mathcal{T}_{m,l}$
⑤Upload channel-wise weights, truncated gradients, and entropy
Client 1
Client M
①
⑤
Server
L layers
⑥Layer-wise weighted aggregation
Layer l
$(\nabla\Theta_{1,l}, e_{1,l})\cdots$
$(\nabla\Theta_{m,l}=$
-1
×
$, e_{m,l})$
$\cdots(\nabla\Theta_{M,l}, e_{M,l})$
Update global model: $\Theta_{g,l}^{t+1}=\Theta_{g,l}^{t}-\nabla\Theta_{g,l}, \nabla\Theta_{g,l}=\sum_{m=1}^{M}p_{m,l}\nabla\Theta_{m,l}$
①Distribute global model Θ_g^t to clients and conduct local model training

System Design: Overview

5. Client-to-Server Transmission

Clients send truncated gradients, weights, and entropy values to the server.

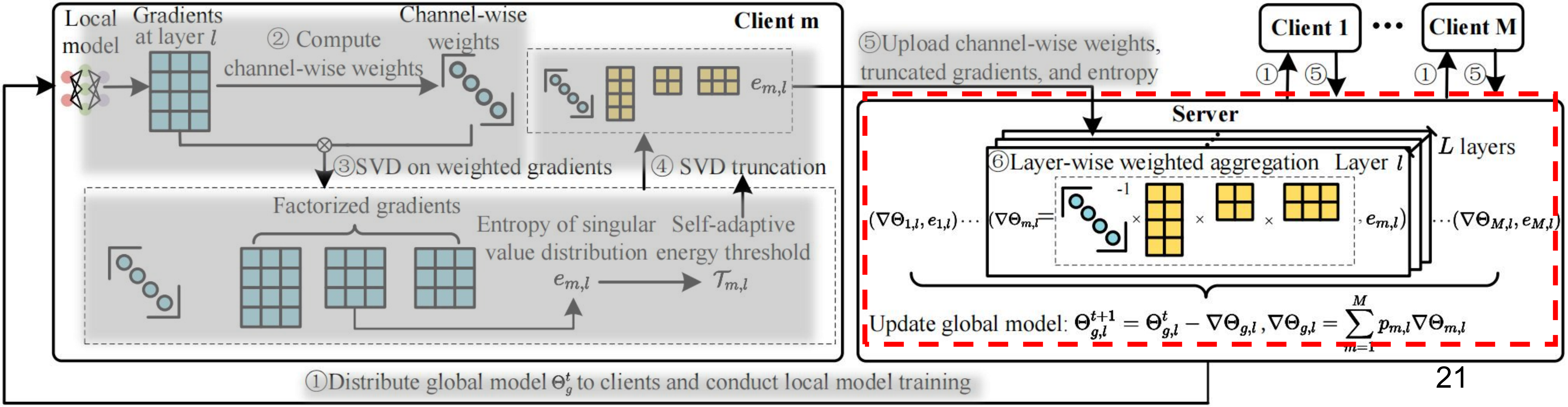


Local model
Gradients at layer l
② Compute channel-wise weights
Channel-wise weights
Client m
$e_{m,l}$
③SVD on weighted gradients
④ SVD truncation
Factorized gradients
Entropy of singular value distribution
Self-adaptive energy threshold
$e_{m,l}$
$\mathcal{T}_{m,l}$
⑤Upload channel-wise weights, truncated gradients, and entropy
Client 1
•••
Client M
①
⑤
①
⑤
Server
L layers
⑥Layer-wise weighted aggregation
Layer l
$(\nabla\Theta_{1,l}, e_{1,l})\cdots$
$(\nabla\Theta_{m,l}=$
-1
×
×
×
$, e_{m,l})$
$\cdots(\nabla\Theta_{M,l}, e_{M,l})$
Update global model: $\Theta_{g,l}^{t+1} = \Theta_{g,l}^{t} - \nabla\Theta_{g,l}, \nabla\Theta_{g,l} = \sum_{m=1}^{M} p_{m,l}\nabla\Theta_{m,l}$
①Distribute global model Θ_g^t to clients and conduct local model training

System Design: Overview

6. Server Aggregation

Server reconstructs gradients, computes aggregation weights, and updates the global model.


Local model
Gradients at layer l
② Compute channel-wise weights
Channel-wise weights
Client m
$e_{m,l}$
③SVD on weighted gradients
④ SVD truncation
Factorized gradients
Entropy of singular value distribution
Self-adaptive energy threshold
$e_{m,l}$
$\mathcal{T}_{m,l}$
⑤Upload channel-wise weights, truncated gradients, and entropy
Client 1
•••
Client M
①
⑤
①
⑤
Server
L layers
⑥Layer-wise weighted aggregation
Layer l
$(\nabla\Theta_{1,l}, e_{1,l})\cdots$
$(\nabla\Theta_{m,l} =$
-1
×
×
×
$, e_{m,l})$
$\cdots(\nabla\Theta_{M,l}, e_{M,l})$
Update global model: $\Theta_{g,l}^{t+1} = \Theta_{g,l}^{t} - \nabla\Theta_{g,l}, \nabla\Theta_{g,l} = \sum_{m=1}^{M} p_{m,l}\nabla\Theta_{m,l}$
①Distribute global model Θ_g^t to clients and conduct local model training

Self-Adaptive Energy Threshold

For challenge C1: Adaptive Thresholding

•Entropy of squared singular value increases with the class balance ratio

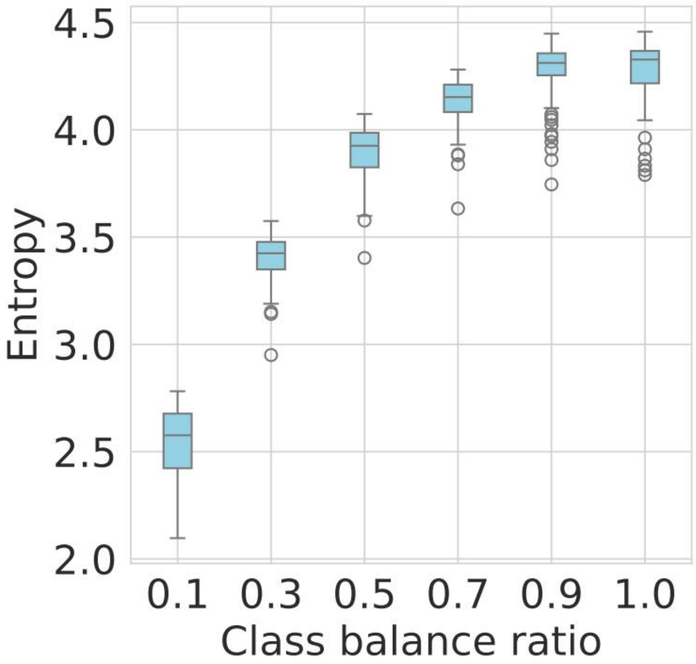
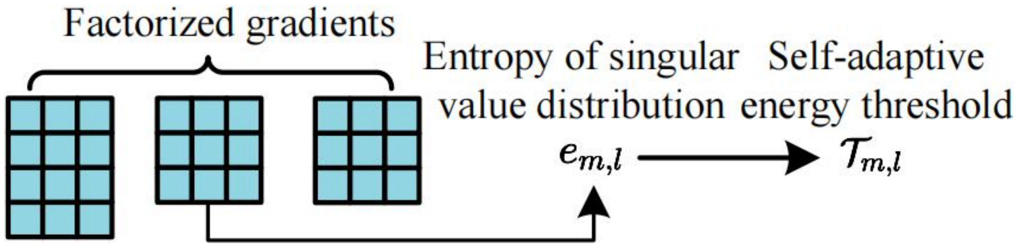


Fig. 1 Entropy of singular value distribution vs. class balance ratio.

•Adapt energy threshold based on entropy values



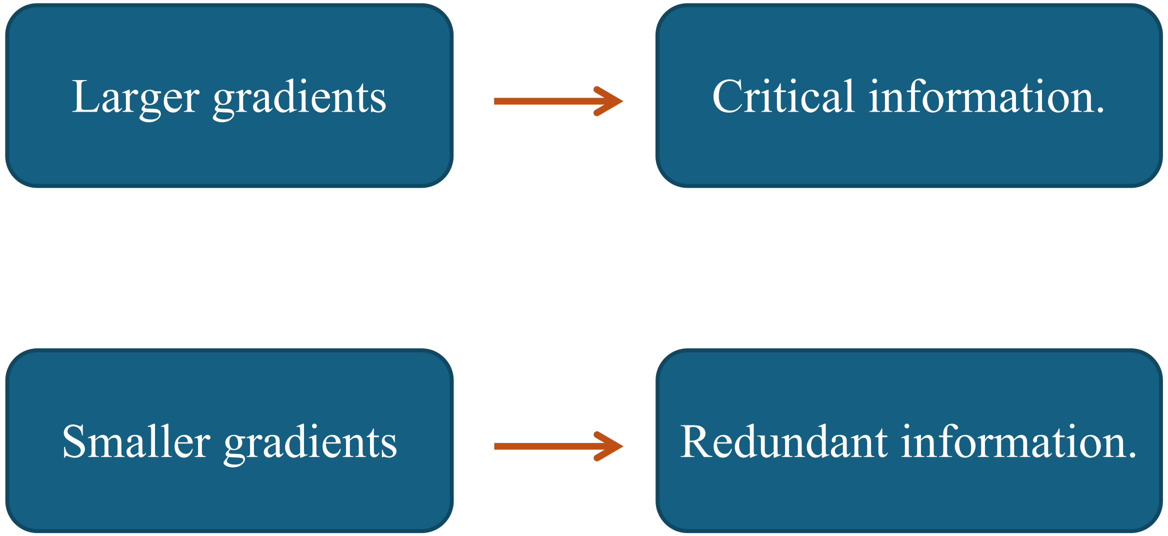
Factorized gradients
Entropy of singular value distribution
Self-adaptive energy threshold
$e_{m,l}$
$\mathcal{T}_{m,l}$

$$\mathcal{T}_{m,l} = 1 - \exp(-\beta e_{m,l})$$

where m is the m − th client, l is the l − th layer. β is the sensitivity parameter.

Channel-Wise Weighted Approximation

For challenge C2: Privacy-Utility Trade-off



Larger gradients → Critical information.

Smaller gradients → Redundant information.

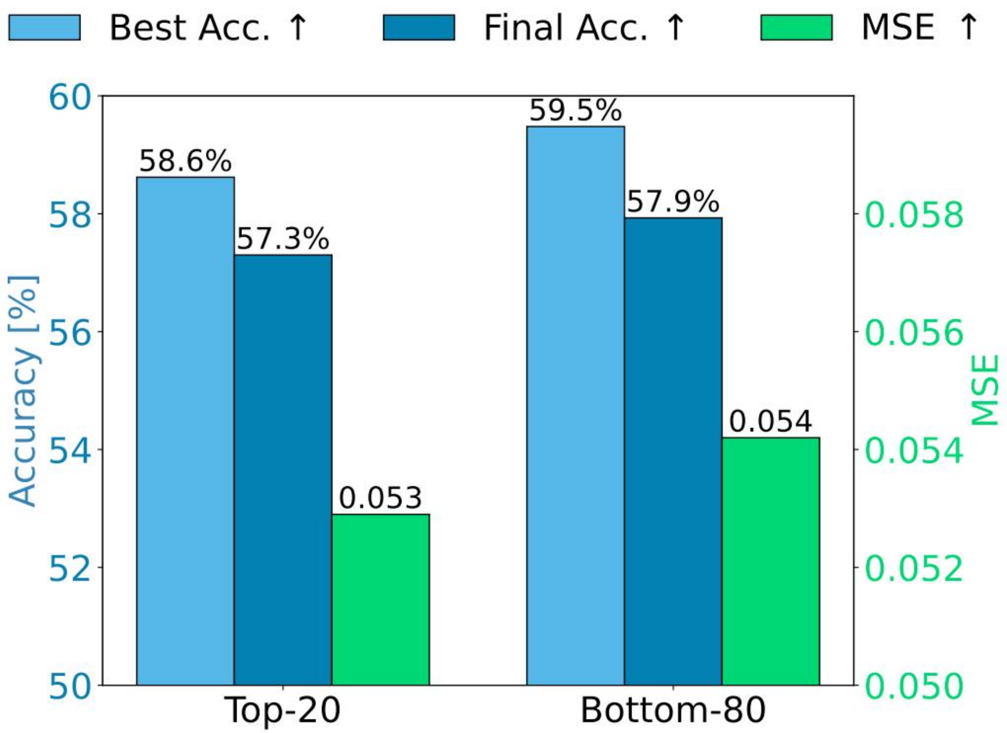
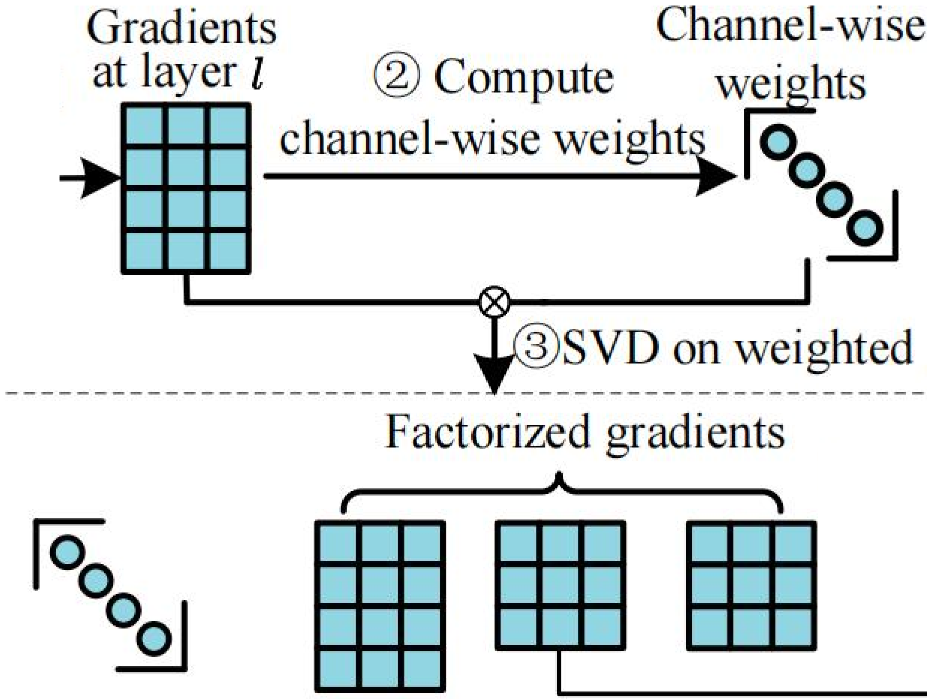


Fig. 1 Comparison of different gradient perturbation-based defense strategies under GIAs.

Channel-Wise Weighted Approximation

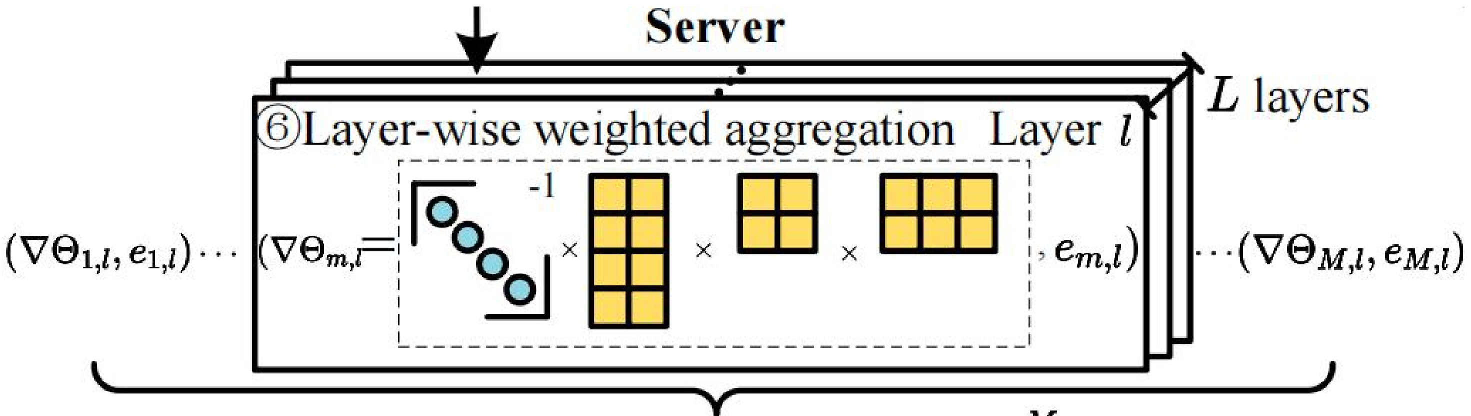
- Preserve larger gradients
- Apply stronger perturbations to smaller gradients.



Layer-Wise Weighted Aggregation

For challenge C3: Robust Aggregation

- Assign higher aggregation weights to clients with more balanced data distribution.



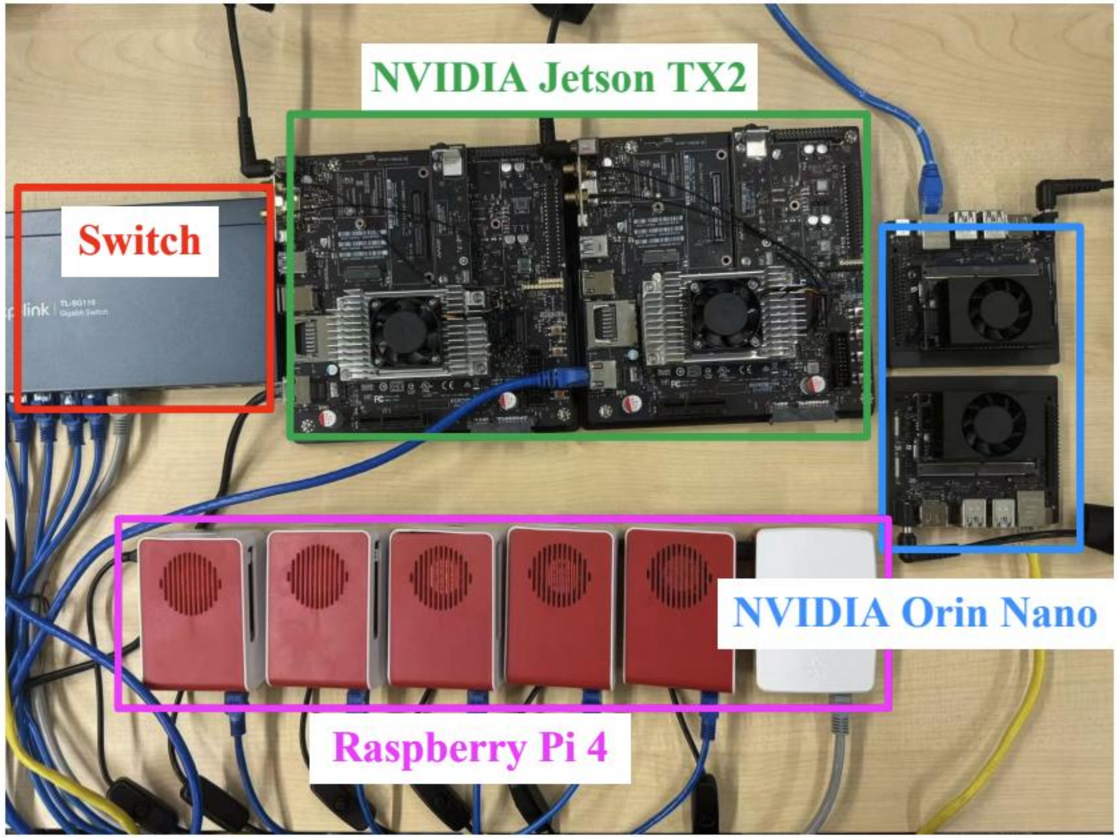
Server
L layers
⑥Layer-wise weighted aggregation Layer l
$(\nabla\Theta_{1,l}, e_{1,l})\cdots$
$(\nabla\Theta_{m,l} =$
-1
×
×
×
$, e_{m,l})$
$\cdots(\nabla\Theta_{M,l}, e_{M,l})$

Update global model: $\Theta_{g,l}^{t+1} = \Theta_{g,l}^{t} - \nabla\Theta_{g,l}\,, \nabla\Theta_{g,l} = \sum_{m=1}^{M} p_{m,l}\nabla\Theta_{m,l}$

Evaluation Applications and Datasets

Application	Dataset	Model
Image Classification (IC-EMNIST)	EMNIST	ResNet-18
Image Classification (IC-CIFAR10)	CIFAR-10	ResNet-18
Human Activity Recognition (HAR)	IMU Dataset	1D ConvNet
Keyword Spotting (KWS)	Google Speech Commands	ResNet-18

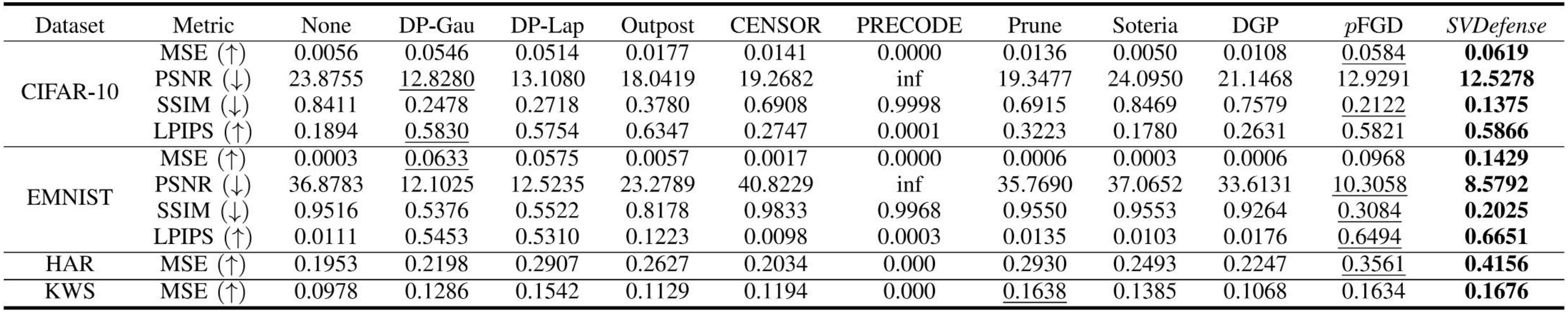
Evaluation Setup



- FL testbed
  - Server
    - AMD EPYC 7543@ 3.7GHz, 256G RAM, and 4 RTX A5000 GPUs.
  - Client devices
    - Two NVIDIA Jetson TX2
    - Two NVIDIA Jetson Nano
    - Six Raspberry Pi 4

Defense Performance

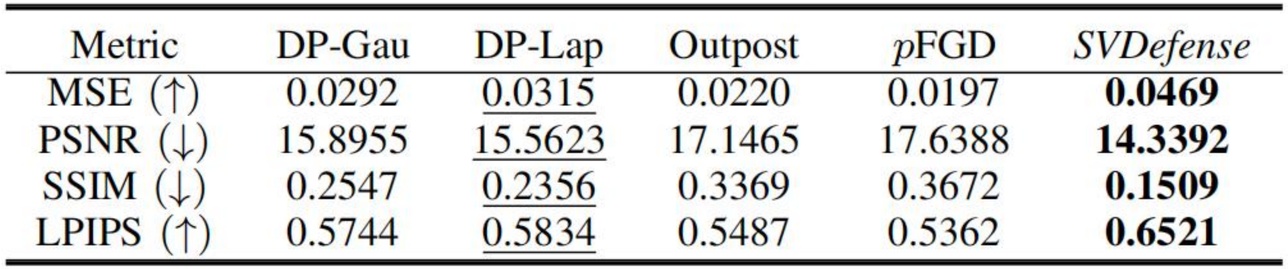
Table: Comparison of Defense Effectiveness Across Different Defense Methods under adaptive IG Attack.



Dataset	Metric	None	DP-Gau	DP-Lap	Outpost	CENSOR	PRECODE	Prune	Soteria	DGP	*p*FGD	*SVDefense*
CIFAR-10	MSE (↑)	0.0056	0.0546	0.0514	0.0177	0.0141	0.0000	0.0136	0.0050	0.0108	0.0584	**0.0619**
	PSNR (↓)	23.8755	12.8280	13.1080	18.0419	19.2682	inf	19.3477	24.0950	21.1468	12.9291	**12.5278**
	SSIM (↓)	0.8411	0.2478	0.2718	0.3780	0.6908	0.9998	0.6915	0.8469	0.7579	0.2122	**0.1375**
	LPIPS (↑)	0.1894	0.5830	0.5754	0.6347	0.2747	0.0001	0.3223	0.1780	0.2631	0.5821	**0.5866**
EMNIST	MSE (↑)	0.0003	0.0633	0.0575	0.0057	0.0017	0.0000	0.0006	0.0003	0.0006	0.0968	**0.1429**
	PSNR (↓)	36.8783	12.1025	12.5235	23.2789	40.8229	inf	35.7690	37.0652	33.6131	10.3058	**8.5792**
	SSIM (↓)	0.9516	0.5376	0.5522	0.8178	0.9833	0.9968	0.9550	0.9553	0.9264	0.3084	**0.2025**
	LPIPS (↑)	0.0111	0.5453	0.5310	0.1223	0.0098	0.0003	0.0135	0.0103	0.0176	0.6494	**0.6651**
HAR	MSE (↑)	0.1953	0.2198	0.2907	0.2627	0.2034	0.000	0.2930	0.2493	0.2247	0.3561	**0.4156**
KWS	MSE (↑)	0.0978	0.1286	0.1542	0.1129	0.1194	0.000	0.1638	0.1385	0.1068	0.1634	**0.1676**

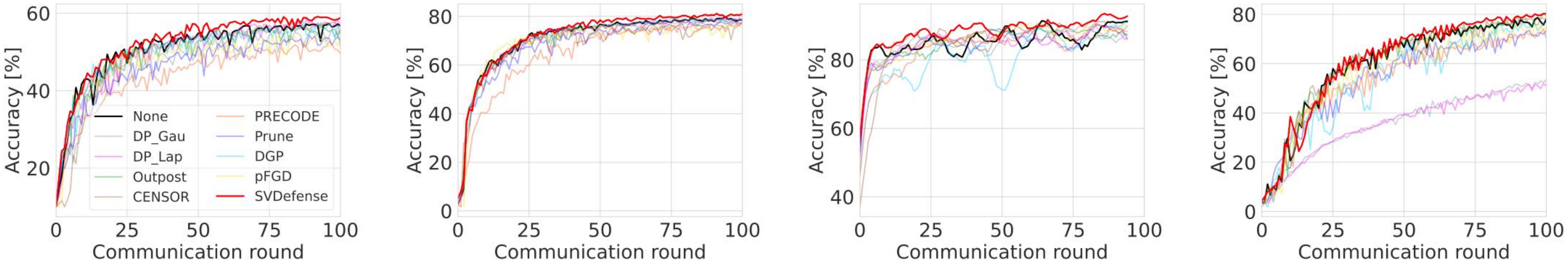
Defense Performance

Table: Comparison of Defense Effectiveness Across Different Defense Methods Under Strong Adaptive LTI Attack.



Metric	DP-Gau	DP-Lap	Outpost	*p*FGD	*SVDefense*
MSE (↑)	0.0292	0.0315	0.0220	0.0197	**0.0469**
PSNR (↓)	15.8955	15.5623	17.1465	17.6388	**14.3392**
SSIM (↓)	0.2547	0.2356	0.3369	0.3672	**0.1509**
LPIPS (↑)	0.5744	0.5834	0.5487	0.5362	**0.6521**

Accuracy Performance



(a) IC-CIFAR10 (b) IC-EMNIST (c) HAR (d) KWS

Figure: Comparison of classification accuracy across different defense methods.

Ablation Study

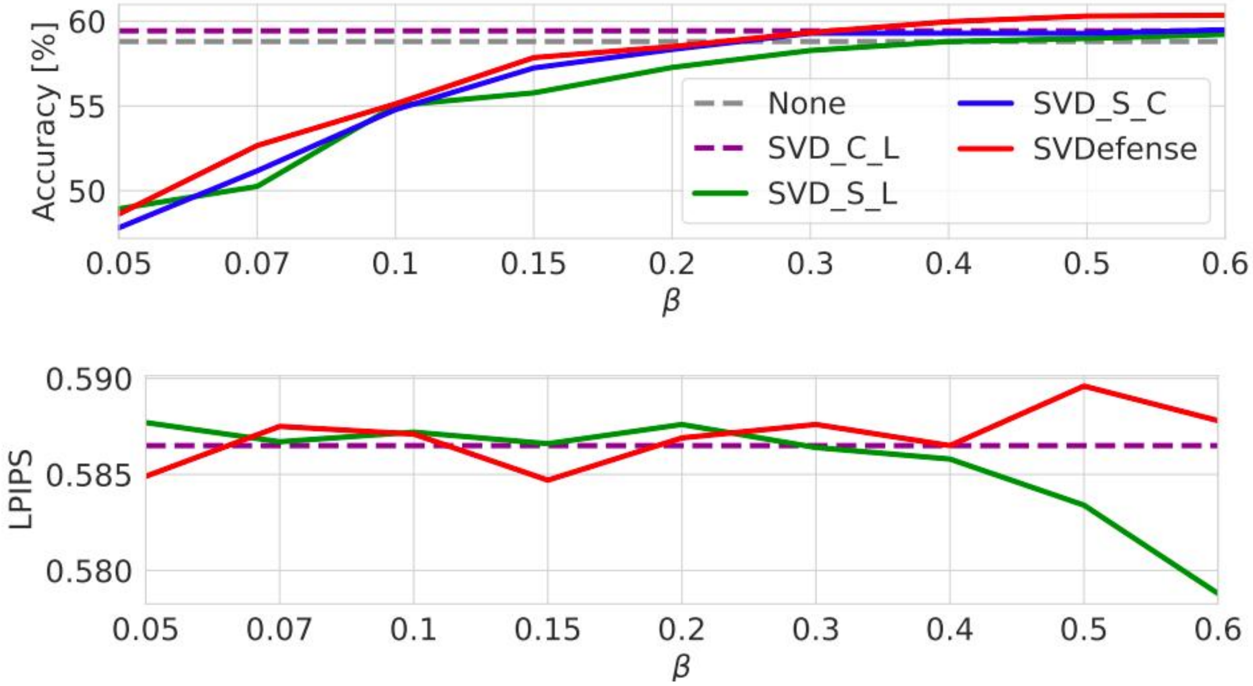


Figure 1: Impact of varying β on accuracy and defense performance for SVDefense.

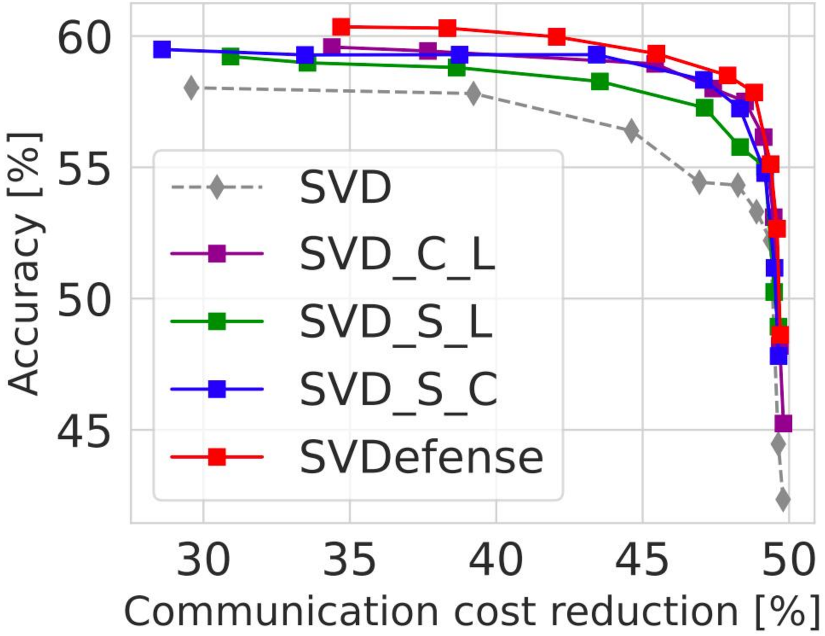


Figure 2: Classification accuracy vs. communication cost reduction.

Latency Performance

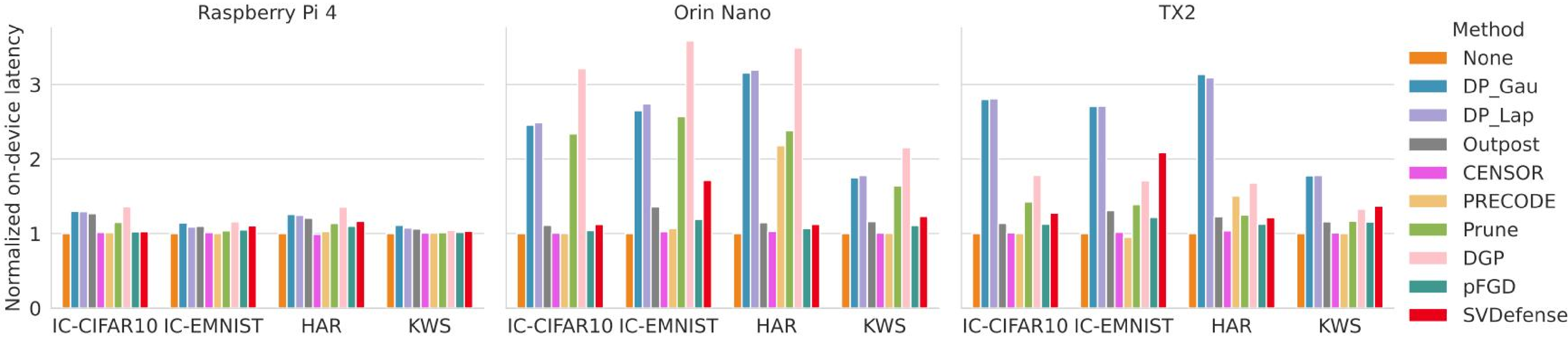


Figure: Comparison of normalized on-device latency across different defense methods on three embedded platforms.

Conclusion

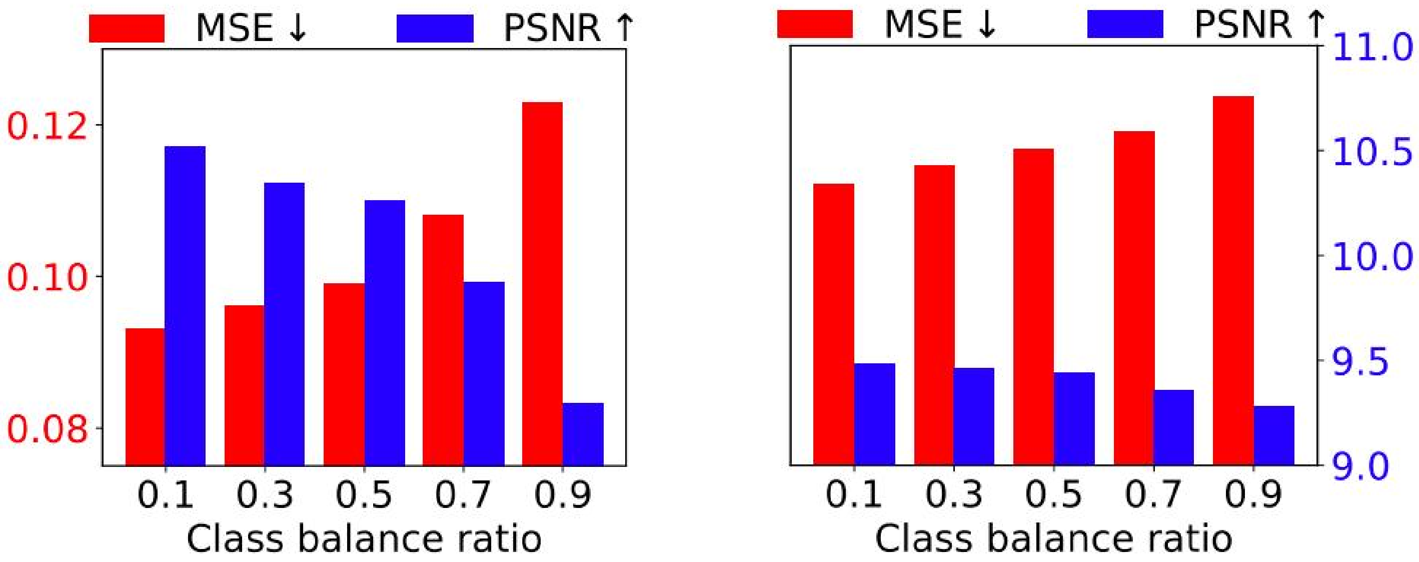
1. Adaptive Attack
We demonstrate the vulnerability of existing defenses to practical adaptive attacks.

2. SVDefense
We propose a novel truncated SVD-based defense against adaptive GIAs in FL.

3. Self-Adaptive Protection
We dynamically adjust protection based on class imbalance.

Thank you!

Evaluation: Adaptive Protection



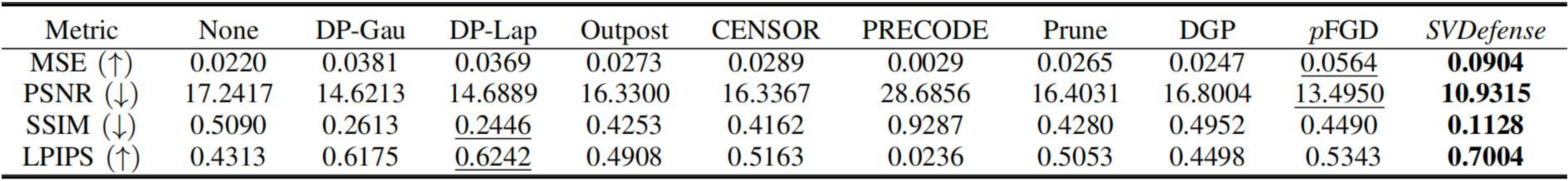
(a) SVD

(b) SVD_S

Figure: Impact of Self-Adaptive Energy Threshold on defense performance under class imbalance.

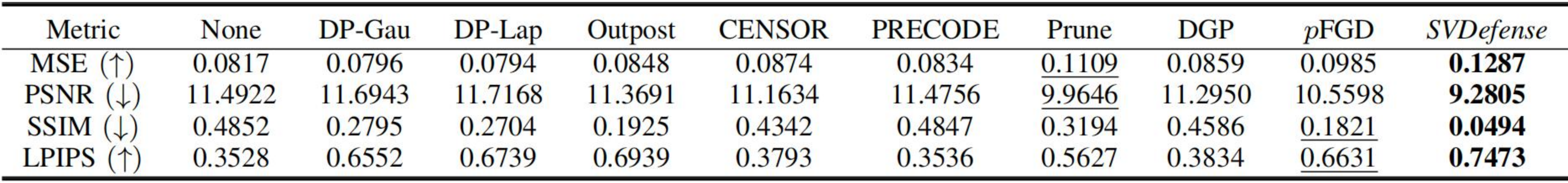
Defense Performance

Table 1: Comparison of Defense Effectiveness Across Different Defense Methods on High-resolution ImageNet with LeNet



Metric	None	DP-Gau	DP-Lap	Outpost	CENSOR	PRECODE	Prune	DGP	*p*FGD	*SVDefense*
MSE (↑)	0.0220	0.0381	0.0369	0.0273	0.0289	0.0029	0.0265	0.0247	0.0564	**0.0904**
PSNR (↓)	17.2417	14.6213	14.6889	16.3300	16.3367	28.6856	16.4031	16.8004	13.4950	**10.9315**
SSIM (↓)	0.5090	0.2613	0.2446	0.4253	0.4162	0.9287	0.4280	0.4952	0.4490	**0.1128**
LPIPS (↑)	0.4313	0.6175	0.6242	0.4908	0.5163	0.0236	0.5053	0.4498	0.5343	**0.7004**

Table 2: Comparison of Defense Effectiveness Across Different Defense Methods on High-resolution ImageNet with ViT



Metric	None	DP-Gau	DP-Lap	Outpost	CENSOR	PRECODE	Prune	DGP	*p*FGD	*SVDefense*
MSE (↑)	0.0817	0.0796	0.0794	0.0848	0.0874	0.0834	0.1109	0.0859	0.0985	**0.1287**
PSNR (↓)	11.4922	11.6943	11.7168	11.3691	11.1634	11.4756	9.9646	11.2950	10.5598	**9.2805**
SSIM (↓)	0.4852	0.2795	0.2704	0.1925	0.4342	0.4847	0.3194	0.4586	0.1821	**0.0494**
LPIPS (↑)	0.3528	0.6552	0.6739	0.6939	0.3793	0.3536	0.5627	0.3834	0.6631	**0.7473**