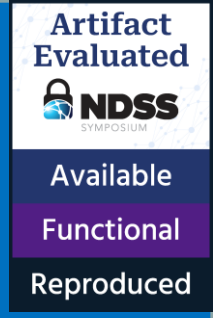


CryptPEFT: Efficient and Private Neural Network Inference via Parameter-Efficient Fine-Tuning

Saisai Xia, Wenhao Wang, Zihao Wang, Yuhui Zhang, Yier Jin, Dan Meng, Rui Hou

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

中国科学院大学
University of Chinese Academy of Sciences

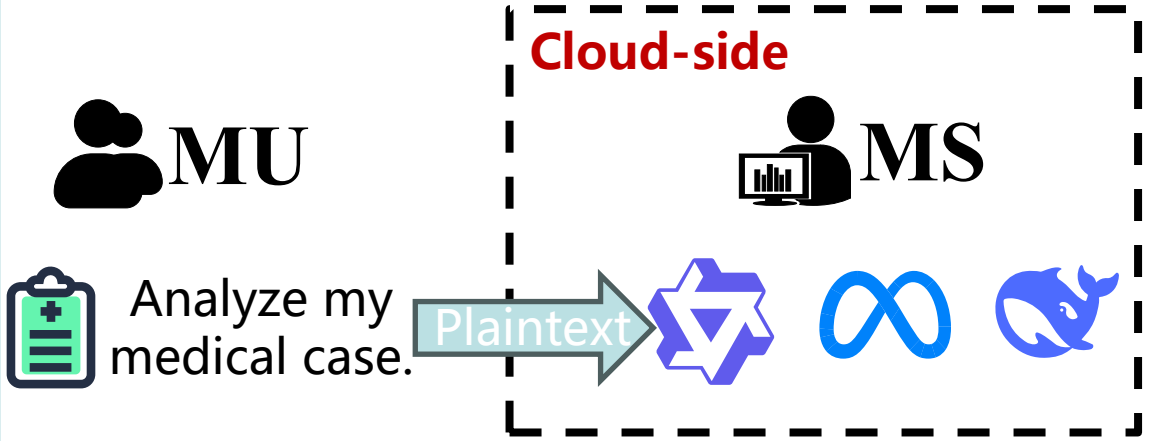
NANYANG
TECHNOLOGICAL
UNIVERSITY
SINGAPORE

1958
University of Science and Technology of China

Privacy Challenges in Neural Network Inference Services

Two deployment paradigms, two fundamental privacy risks.

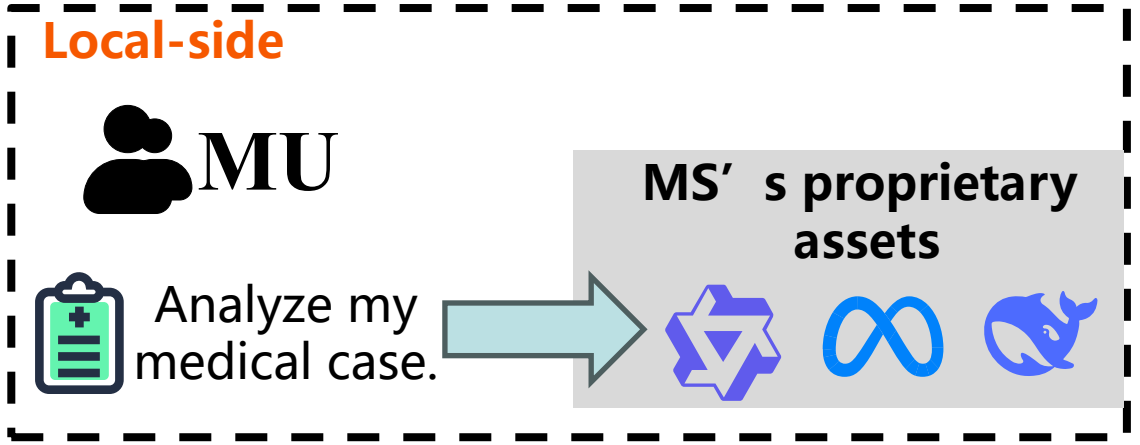
Cloud Inference



Private input leakage

What if my sensitive information is excessively collected?

Local Inference



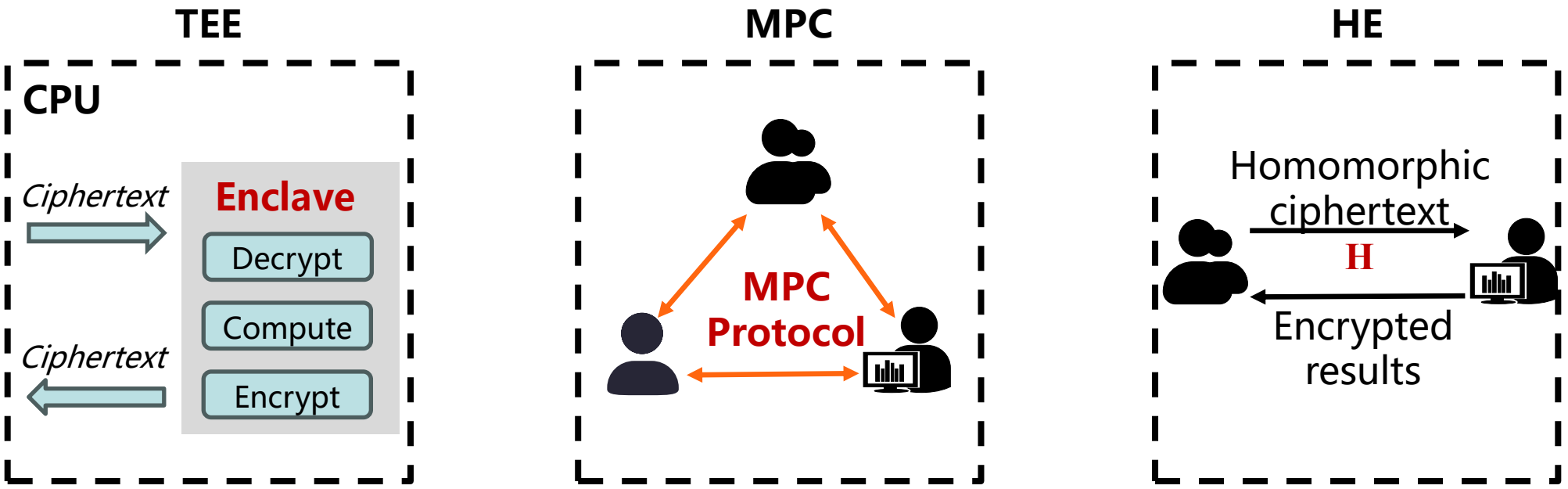
Model IP leakage

What if an attacker tries to steal the model' s parameters, architecture, or training data?

Cloud threatens data privacy; local deployment exposes model IP.

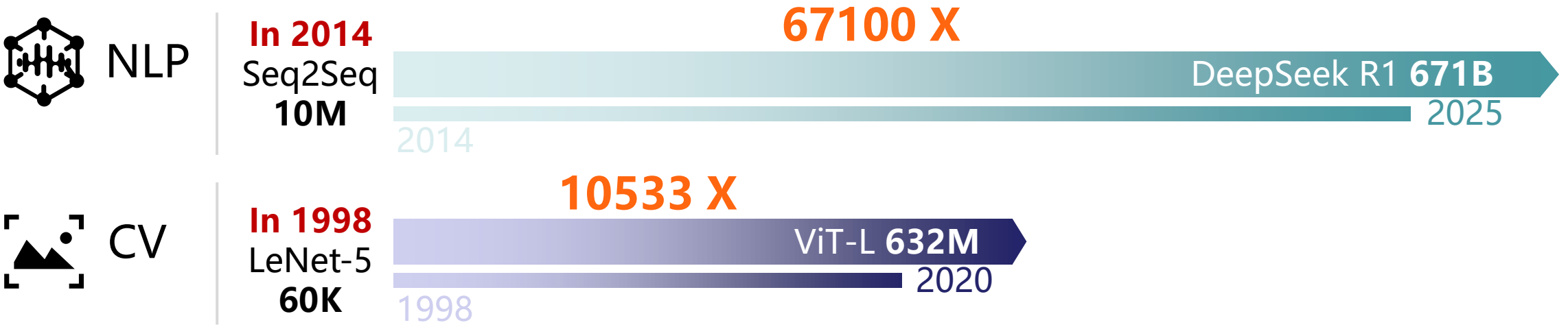
Privacy-preserving Computing Solutions

Technique	Trust Basis	Performance	Security
TEE	Hardware vendor	High	Relies on hardware security
MPC	Distributed protocols	Moderate	Strong
HE	Hard mathematical problems (lattices)	Low	Strong



Neural Network Models Are Becoming Increasingly Large

Four orders of magnitude in NLP & CV growth

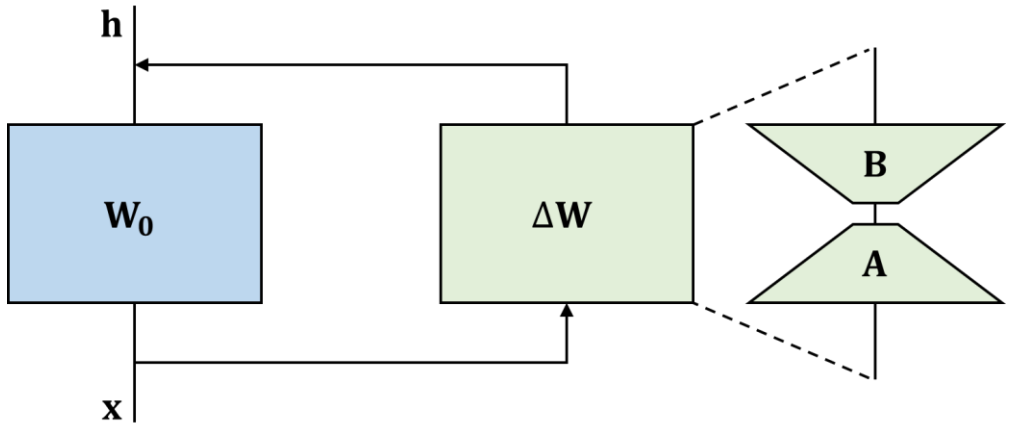


- TEEs have limited compute capability and memory, making them unsuitable for large-scale models.
- HE is computationally expensive and incurs high latency.
- MPC is somewhat faster but requires heavy communication, leading to high latency.

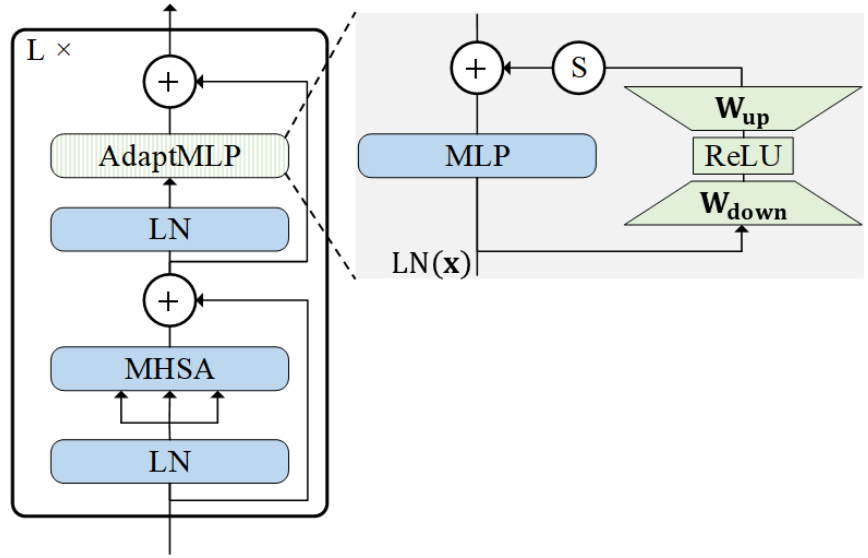
Parameter-Efficient Fine-Tuning

- Only a small set of trainable parameters is added instead of fine-tuning the entire model.
- High parameter efficiency (0.1–10% trainable parameters with comparable or better accuracy), reduced training cost, low storage overhead (shared backbone), strong transferability (multi-task and continual learning), mitigated catastrophic forgetting (frozen base model), and deployment-friendly.

LoRA

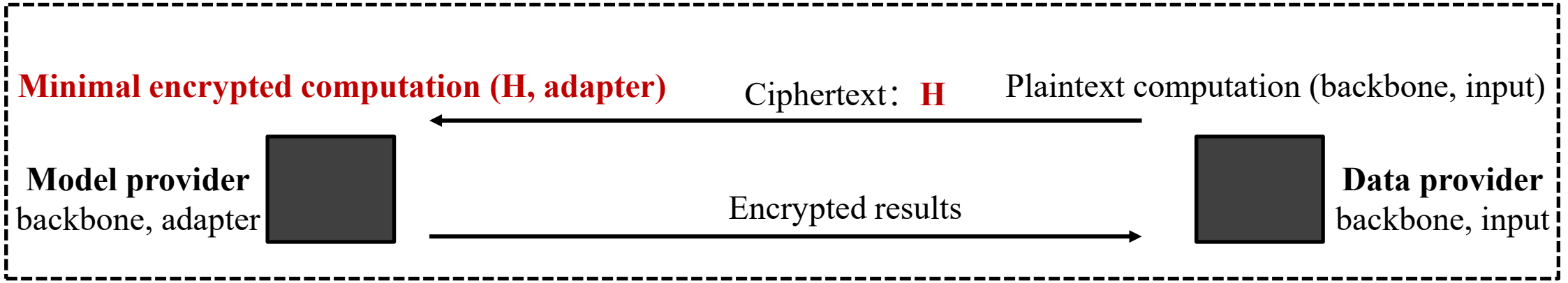


AdaptFormer



Can PEFT Be Used to Accelerate Private Inference?

- **Idea**: Keep the backbone public and maximize plaintext computation to minimize encrypted workloads.

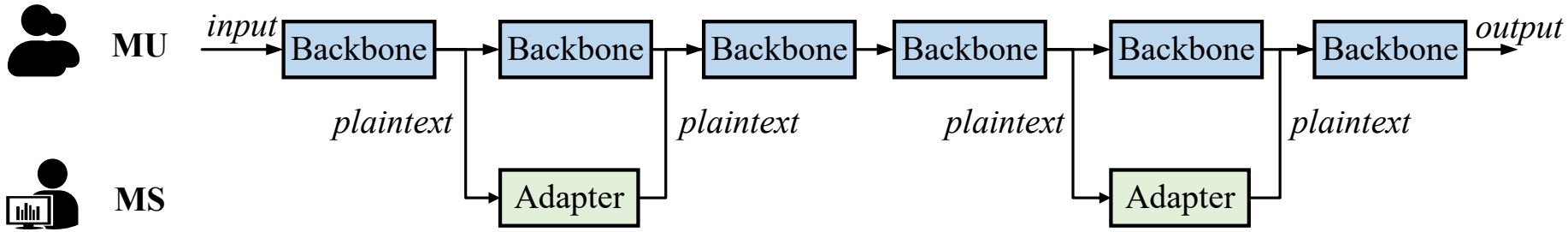


Ideal

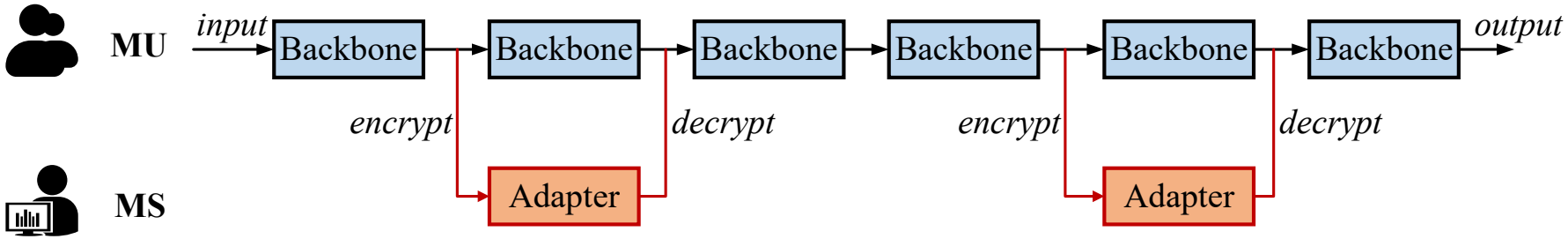
- **Reality**: Mainstream backbone–adapter designs structurally conflict with private inference.

- **Reason**: Once adapters process encrypted data, their outputs "contaminate" the backbone, forcing subsequent computation to remain encrypted.

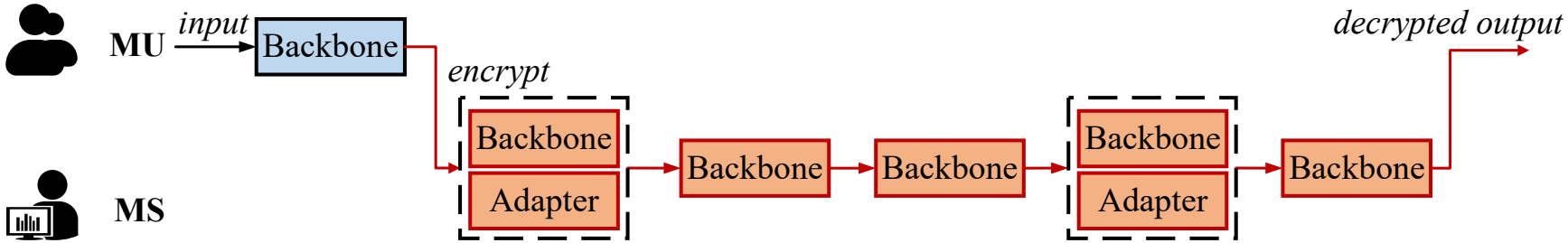
Structural Tension Between PEFT and Private Inference



(a) Plaintext inference



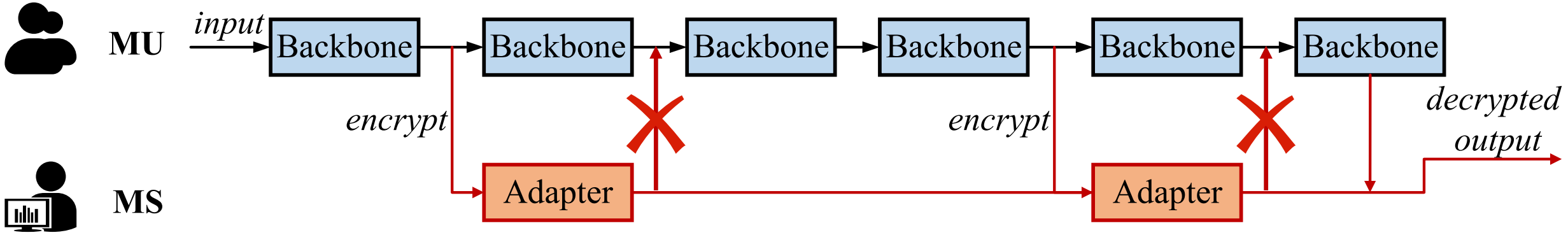
(b) Encrypt-then-decrypt inference (insecure)



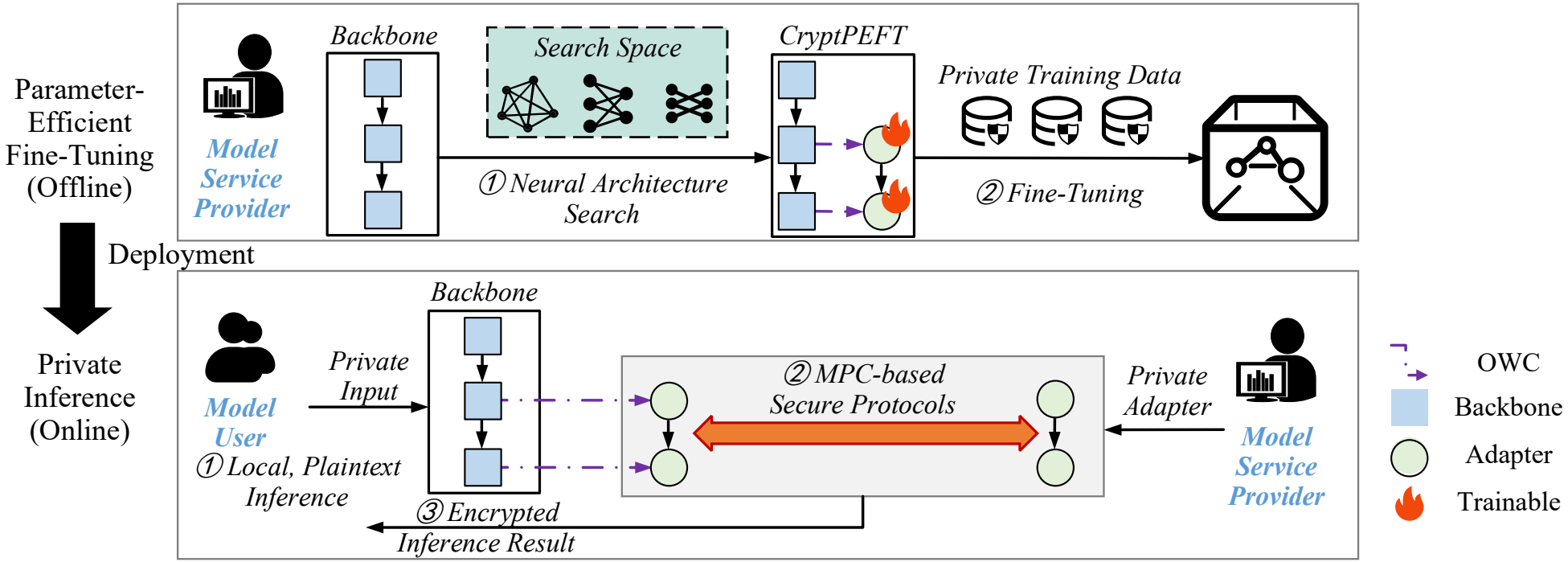
(c) Private inference (secure but slow)

Solution: A PEFT Architecture for Private Inference — CryptPEFT

- Solution: One-way communication，OWC
  - **One-way data flow**
  - "**No return path**"



Workflow of CryptPEFT



Offline

① The Model Service Provider uses NAS to search for the target adapter.

② The Model Service Provider fine-tunes the target adapter using its private data.

Online

① The Model User runs local inference on the backbone in plaintext.

② The Model User encrypts intermediate backbone outputs and, with the Model Service Provider, runs private inference on the adapter via MPC.

③ The Model Service Provider returns the encrypted results to the Model User.

OWC-compatible Adapter Design

- **Challenge**: **Designing OWC-compliant adapters**
  - Balance utility and private-inference efficiency
  - Adapt to diverse downstream tasks
  - Make NAS more efficient

- Understand OWC’ s impact on model utility (accuracy)
  - Adapter architecture
  - Adapter placement
  - NAS: search for optimal adapter configurations

- Understand how operators affect private-inference efficiency
  - Softmax
  - GELU
  - Linear

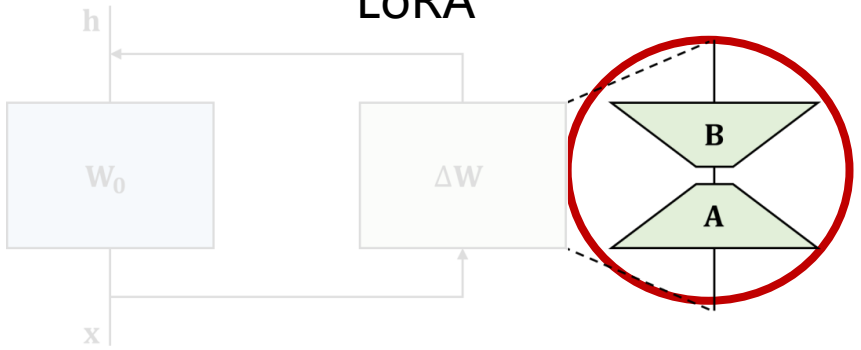
Do Traditional Adapters Work for CryptPEFT?

- Replace standard PEFT's two-way communication (TWC) with one-way communication (OWC).
  - Reuse traditional adapters
  - Measure model utility across tasks

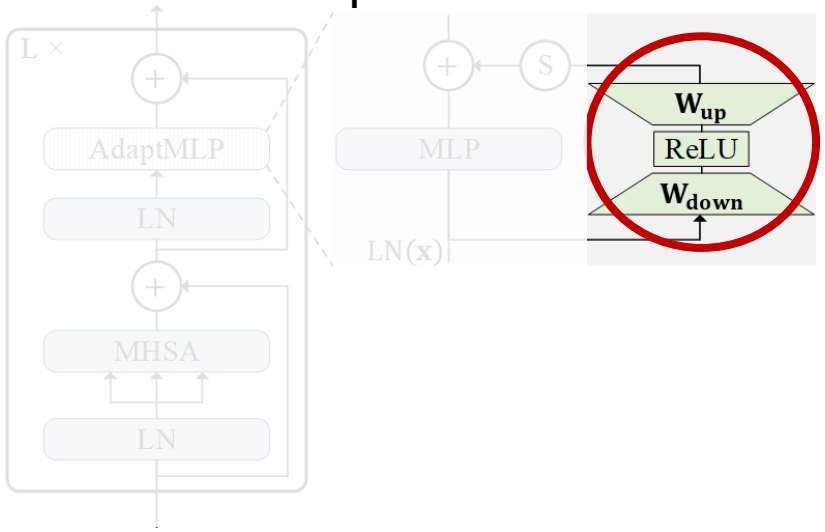
Datasets	LoRA		AdaptFormer	
	TWC	OWC	TWC	OWC
CIFAR-10	97.31	95.84 (↓1.47)	97.34	95.91 (↓1.43)
CIFAR-100	87.41	83.73 (↓3.68)	87.23	83.72 (↓3.51)
Food-101	83.95	80.13 (↓3.82)	83.91	80.02 (↓3.89)
SVHN	91.81	63.17 (↓28.64)	91.72	63.11 (↓28.61)
Flowers-102	84.37	80.76 (↓3.61)	84.42	80.89 (↓3.53)

- Directly migrating traditional adapters to CryptPEFT degrades performance.
- Traditional adapters lack attention modules and cannot meet CryptPEFT's requirements.

LoRA



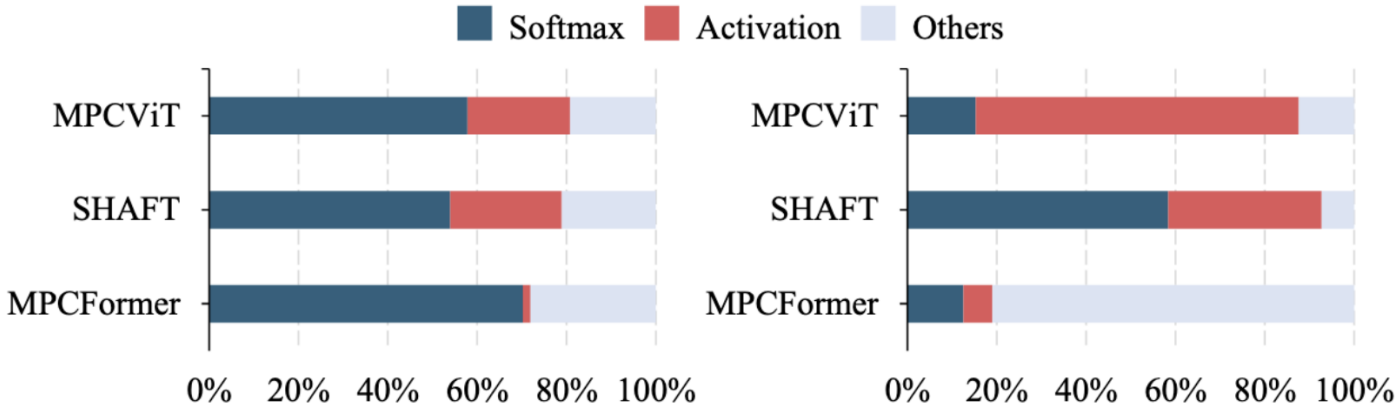
AdaptFormer



New adapters tailored for CryptPEFT are required.

Are Existing Attention Efficient for Private Inference?

- Measure the communication overhead of different attention mechanisms in private inference.
  - Communication Round、Communication Cost



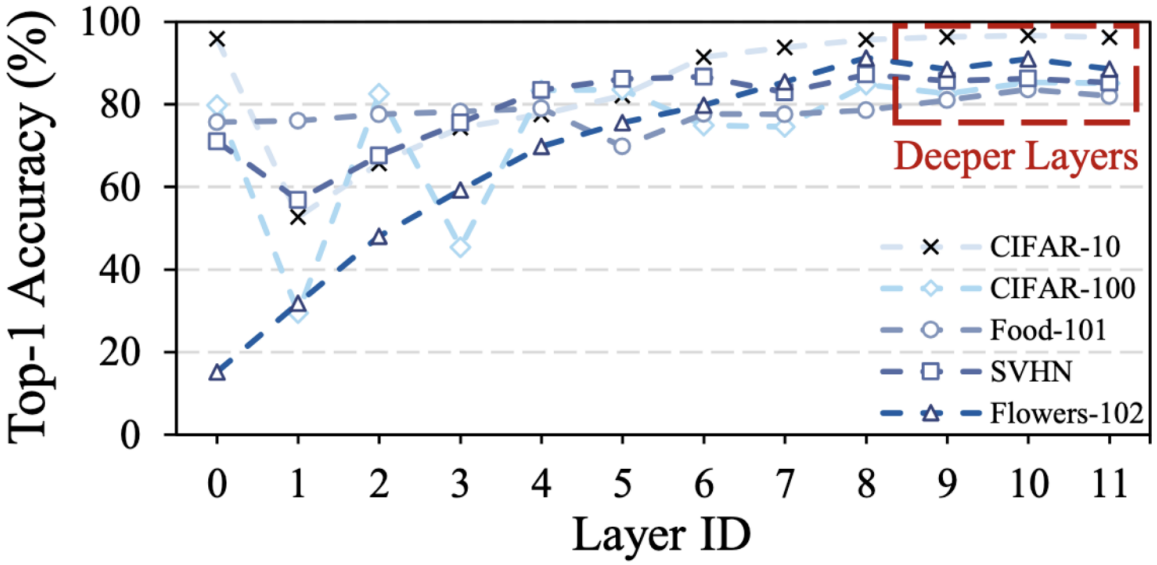
(a) Communication Round (b) Communication Cost

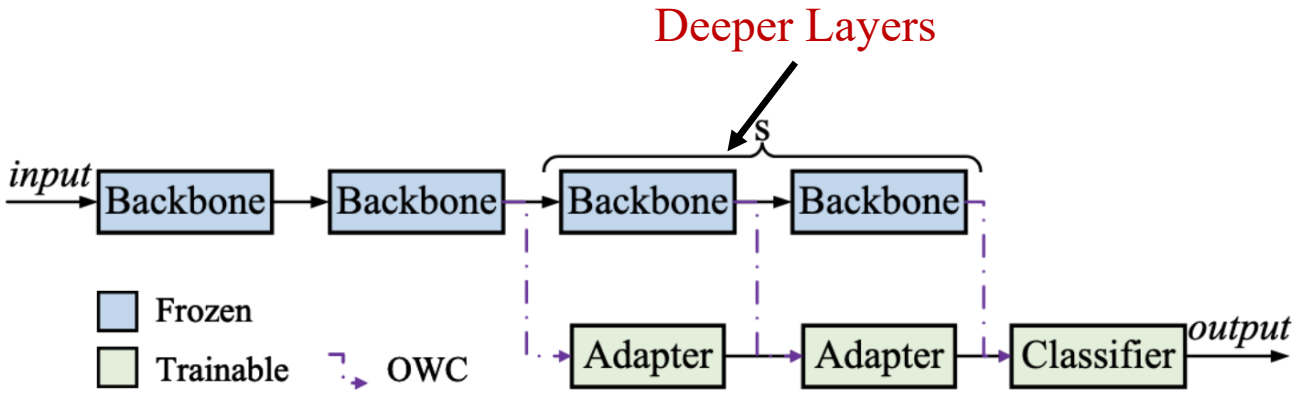
- Existing approaches to accelerating attention in private inference remain inefficient.
- In MPCViT and SHAFT, softmax and activations account for over 80% of total communication.
- MPCFormer reduces bandwidth usage but still suffers from excessive communication rounds.

More communication-efficient designs are needed to replace current attention.

Where Should Adapters Be Placed?

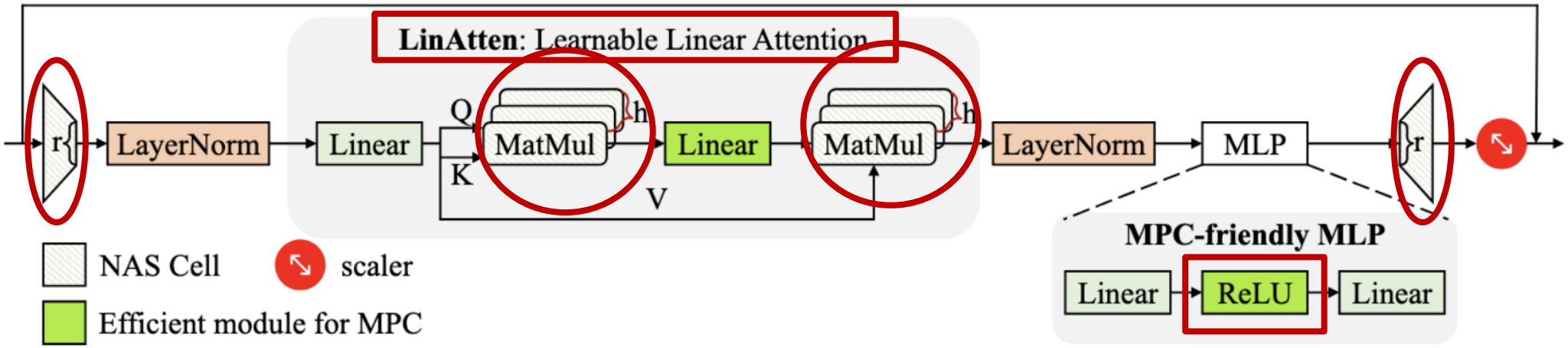
- Under OWC: insert a single adapter into one backbone layer
  - Record the layer ID
  - Measure task-specific model utility





Adapters should be placed deeper in the backbone.

Adapter Architecture Design



Key Design

- ✓ **LinAtten**: Replace softmax with a learnable linear module.
- ✓ **MPC-friendly MLP**: Replace GELU with communication-efficient ReLU.
- ✓ **LoRA**: Low-rank compression to drastically reduce adapter parameters.
- ✓ **NAS**: Predefine NAS cells to optimize adapter architectures.

CryptPEFT-Aware Neural Architecture Search

Algorithm 1: NAS-based search algorithm.

Input: Targets $\mathcal{U}_{\text{target}}, \mathcal{L}_{\text{target}}, \mathcal{T}_{\text{target}}$; Latency model $\text{Latency}(*,*,*)$; Search spaces $\mathcal{H}, \mathcal{R}, \mathcal{S}$

Output: Best config (h^*, r^*, s^*); Best utility U^*

$(s, \Delta) \leftarrow (1, 1)$;
$(h^*, r^*, s^*, \mathcal{U}^*) \leftarrow (\perp, \perp, \perp, -\infty)$;
while $s \leq max(\mathcal{S})$ **do**
  $\tau \leftarrow 0$;
  while *True* **do**
    $(h, r) \leftarrow \mathcal{C}(\theta)$;
    if $Latency(h, r, s) > Latency(h_{init}, r_{init}, s+\Delta)$ *or* $Latency(h, r, s) > \mathcal{L}_{target}$ **then**
      $(Reward, \tau) \leftarrow (1/\text{Latency}(h, r, s), \tau + 1)$;
    end
    else
      $\mathcal{U} \leftarrow \text{Eval}(h, r, s)$;
      $Reward \leftarrow \mathcal{U} + 1/\text{Latency}(h, r, s)$;
      $\tau \leftarrow \tau + 1$;
      if $\mathcal{U} > \mathcal{U}^*$ **then**
        $(h^*, r^*, s^*, \mathcal{U}^*) \leftarrow (h, r, s, \mathcal{U})$;
        $\tau \leftarrow 0$;
      end
      if $\mathcal{U}^* \geq \mathcal{U}_{target}$ **then return** (h^*, r^*, s^*, U^*);
    end
    $\theta \leftarrow \mathcal{OPT}(\mathcal{C}, \theta, Reward)$;
    if $\tau \geq \mathcal{T}_{target}$ **then break**;
  end
  $s \leftarrow s + 1$;
end
return (h^*, r^*, s^*, U^*);

- Traditional NAS only considers parameter count.
- MPC-based private inference must account for communication volume and rounds.

Key Design

- ✓ **Search strategy**: latency-aware and more efficient than traditional NAS
- ✓ **Search space**: general, effective, and efficient
- ✓ **Latency model**: estimates private-inference time

Latency model——WAN

$$Latency(h, r, s) = (0.03828h + 0.00465r + 0.48139)s + 0.32122$$

Latency model——LAN

$$Latency(h, r, s) = (0.02747h + 0.00296r + 0.12313)s + 0.24229$$

Evaluation

- Implemented with the MPC-based framework CrypTen[1]
- Simulated network settings
  - **WAN**: 400 Mbps bandwidth, 4 ms latency
  - **LAN**: 1 Gbps bandwidth, 0.1 ms latency

- Baseline
  - **Traditional PEFT**: not optimized for encrypted computation
  - **Simple fine-tuning (SFT)**: fine-tune the last one or two layers
  - All baselines adopt SHAFT's approximation methods[2]

[1] Knott B, Venkataraman S, Hannun A, et al. Crypten: Secure multi-party computation meets machine learning[J]. Advances in Neural Information Processing Systems, 2021, 34: 4961-4973.
[2] Kei A Y L, Chow S S M. SHAFT: Secure, Handy, Accurate, and Fast Transformer Inference[C]//Network and Distributed System Security Symposium, NDSS. 2025, 2025.

Model Utility

Methods		Avg. private params. (M)	CIFAR-10	CIFAR-100	Food-101	SVHN	Flowers-102	Avg. utility
PEFT (baseline)	LoRA	86.01	97.31%	**87.41%**	83.95%	91.81%	84.37%	88.97%
	AdaptFormer	87.05	97.34%	87.23%	83.91%	91.72%	84.42%	88.92%
SFT (baseline)	Last Layer	7.15	97.28%	86.24%	84.99%	87.35%	88.81%	88.93%
	Last 2 layers	14.23	**97.51%**	86.57%	**85.56%**	90.10%	90.63%	90.07%
CRYPTPEFT	Utility-first (LAN)	1.28	97.29%	85.63%	84.84%	**91.82%**	**92.60%**	**90.43%**
	Efficiency-first (LAN)	**0.46**	97.19%	85.37%	84.56%	90.03%	91.32%	89.69%
	Utility-first (WAN)	1.53	97.26%	85.70%	84.70%	91.51%	**92.60%**	90.35%
	Efficiency-first (WAN)	0.80	97.23%	85.47%	84.38%	90.27%	91.45%	89.76%

- Utility-first outperforms traditional PEFT by **1.45%.**
- Efficiency-first outperforms SFT by **0.8%**
- **88.81%** fewer trainable parameters than SFT.

Private Inference Latency

	Methods		CIFAR-10	CIFAR-100	Food-101	SVHN	Flowers-102
LAN	**PEFT** (baseline)	LoRA	271.34	270.39	266.42	266.59	267.82
		AdaptFormer	269.02	269.80	270.75	268.31	271.65
	SFT (baseline)	Last layer	23.30	23.56	23.44	23.17	23.72
		Last 2 layers	46.89	46.96	44.95	45.28	47.24
	CRYPTPEFT	Efficiency-first	**0.81** (0.83)	**1.13** (1.10)	**1.04** (1.06)	**0.90** (0.87)	**0.75** (0.75)
WAN	**PEFT** (baseline)	LoRA	548.27	545.91	547.08	546.43	545.56
		AdaptFormer	549.95	551.08	550.64	549.66	548.96
	SFT (baseline)	Last layer	45.32	45.38	45.25	45.56	44.26
		Last 2 layers	90.70	90.80	90.07	88.08	90.54
	CRYPTPEFT	Efficiency-first	**2.45** (2.55)	**2.26** (2.24)	**1.85** (1.79)	**2.78** (2.66)	**1.61** (1.68)

	Metrics	SFT (baseline)	CRYPTPEFT	Improvements
LAN	Comm. (GB)	1.55	0.05	31.00×
	Comm. round	77	29	2.66×
	Comm. time (s)	14.40	0.55	26.18×
	Total time (s)	23.56	1.13	20.85×
WAN	Comm. (GB)	1.55	0.06	25.83×
	Comm. round	77	29	2.66×
	Comm. time (s)	34.42	1.51	22.79×
	Total time (s)	45.38	2.26	20.08×

- Compared to SFT: **20.85×**
- Compared to PEFT: **238.76×**
- Under WAN settings, CIFAR-100
  - latency: **2.26 s**, model utility: **85.47%**

Compared to MPCViT[1]

- Model utility: improves by **2.92% --- 8.01%**
- Inference efficiency: achieves a **9.19× --- 13.14 ×** speedup

Settings		Metrics	MPCViT	CryptPEFT	Improvements
CIFAR-10	LAN	Utility	94.27%	97.19%	↑2.92%
		Total time	10.64	0.81	13.14×
	WAN	Utility	94.27%	97.23%	↑2.96%
		Total time	24.10	2.45	9.84×
CIFAR-100	LAN	Utility	77.46%	85.37%	↑7.91%
		Total time	10.38	1.13	9.19×
	WAN	Utility	77.46%	85.47%	↑8.01%
		Total time	22.97	2.26	10.16×

[1] Zeng W, Li M, Xiong W, et al. Mpcvit: Searching for accurate and efficient mpc-friendly vision transformer with heterogeneous attention[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 5052-5063.

Summary

- PEFT and private inference have a fundamental structural conflict
- Redesign PEFT and adapter architectures for efficient private inference
  - One-way data flow
  - Efficient attention design
  - NAS

- **Future Directions**: TEE-accelerated private LLM inference
  - TEE
  - One-way data flow
  - ……

- Code: https://github.com/Saisai-Xia/CryptPEFT