中国电信云计算研究院
China Telecom Cloud Computing Research Institute

IEEE INFOCOM

Mercury: Towards Optimal Accuracy-Latency Trade-off for Collaborative Transformer Inference

Yumeng Liang[†], Jianhui Chang[†], Sijia Li[‡†], Mingyuan Zang[†], Jie Wu[†*]
[†]China Telecom Cloud Computing Research Institute, China
[‡]University of Science and Technology Beijing, China
[*]Temple University, USA
{liangym9, changjh1}@chinatelecom.cn, lisijia@xs.ustb.edu.cn, zangmy1@chinatelecom.cn, jiewu@temple.edu.

Background: Vision Transformer

- Vision Transformer (ViT): powerful neural networks for vision tasks
  - Superior accuracy performance vs. convolutional neural networks (CNNs)

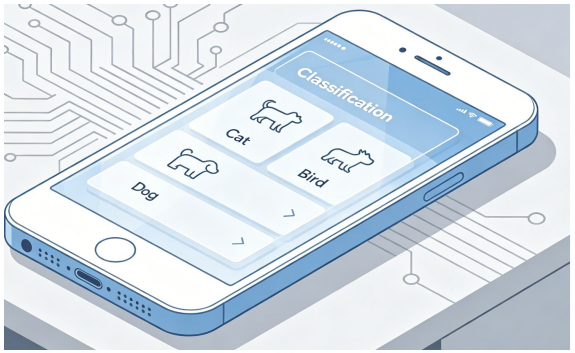
    E.g., ViT-Base achieves 5% higher accuracy compared to ResNet on image classification
  - Yet with higher computation cost (4.3x higher FLOPS)

Method	Accuracy on ImageNet	Computation Cost
ResNet-50	80.4%	4.1 GFLOPs
ViT-Base	85.4% (+5%)	17.5 GFLOPs (4.3x)

Background: Emerging mobile applications

- ViT is promising to provide visual intelligence service in emerging mobile applications like:

Wearable AI



Mobile intelligence

In-car Vision

- However, these scenarios face strict constraints to deploy powerful ViT:
  - Limited hardware computing resources compared to the cloud
  - Strict latency requirements for real-time interaction
  - Tight power budgets for deployment on mobile devices

How to achieve accurate ViT inference with low latency on mobile devices?

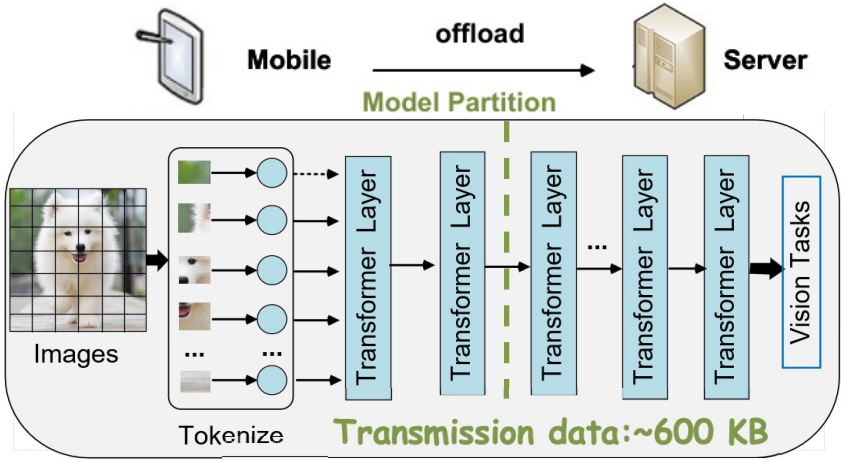
Mobile-Cloud Collaborative Inference

Collaborative Inference: Split the model, and partially offload the workload to the cloud

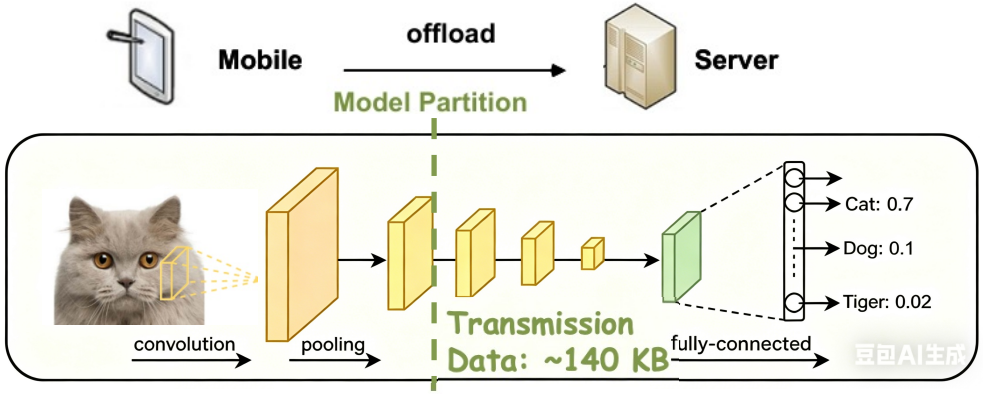
- Advantages: 1. Accelerate the inference speed, leveraging the powerful cloud resources
  2. Reduce power consumption of the mobile, due to released computation

However, ViT has a unique architecture distinct from CNNs:

- Constant intermediate feature size across all layers, instead of the layer-by-layer progressively shrinking one of CNNs, enlarges the transmission latency



ViT architecture

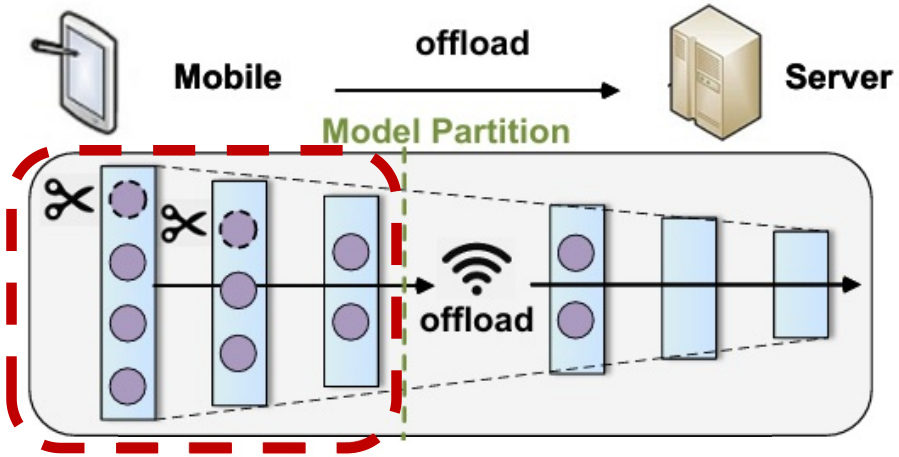


CNN architecture

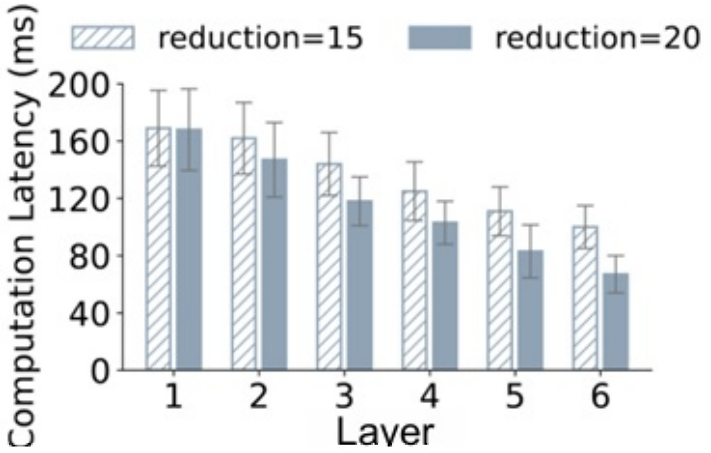
Existing approaches

Existing approaches: Pruning tokens progressively to reduce the transmission latency[1,2]

Advantages: Reducing the transmission overhead when split at deeper layers (like CNNs)



Framework of existing methods



Computation latency of different layers

- However, such layer-by-layer reduction leads to imbalanced workloads:

***Early layers** have **heavier computation**, but have to run on **resource-constrained** mobile devices!*

[1] L. Jiang et al, "Janus: Collaborative vision transformer under dynamic network environment," INFOCOM 2025

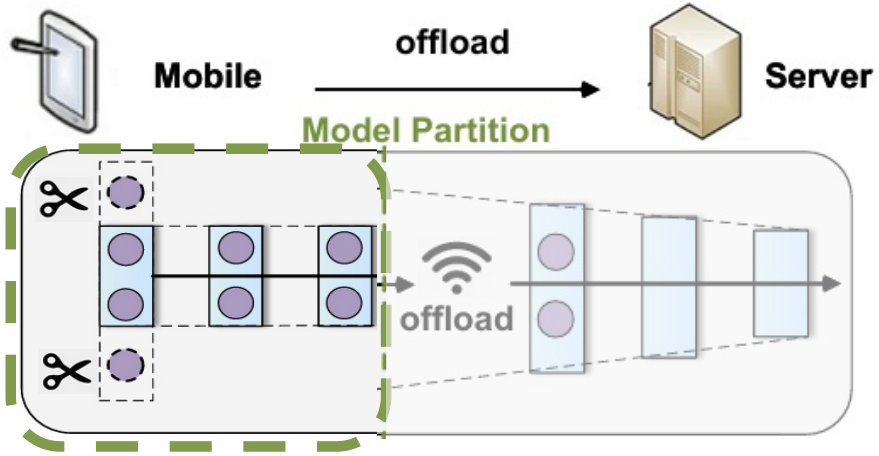
[2] J. Chen et al, "Otas: An elastic transformer serving system via token adaptation," INFOCOM 2024

Opportunities and Challenges

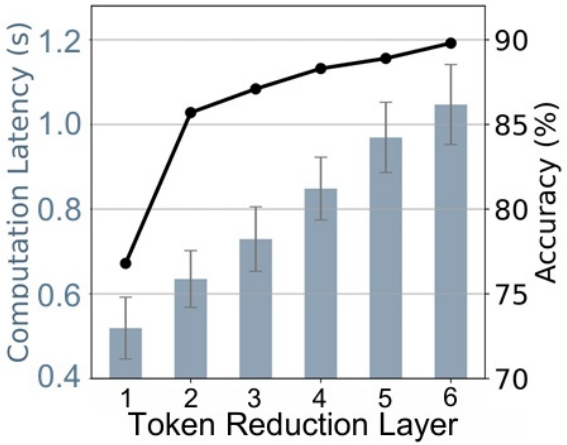
Token Pruning at early layer exhibit extensive computation latency reduction at mobile devices

- E.g., Prune 50% tokens at Layer 1 instead of Layer 6 reduces 50% of computation latency

However, it sacrifices accuracy drastically, which drops from 90% to 77%



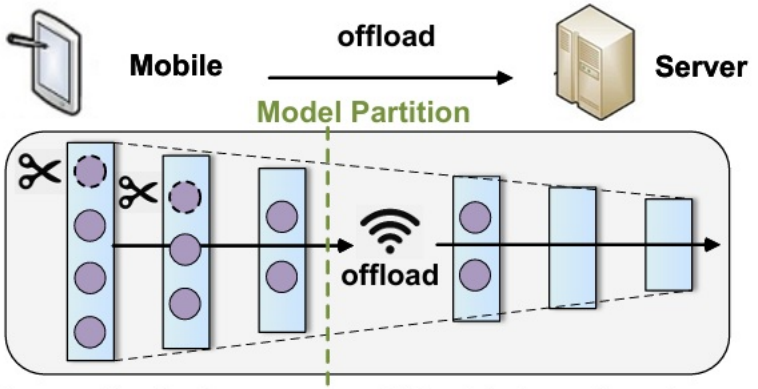
Token pruning at the initial layer



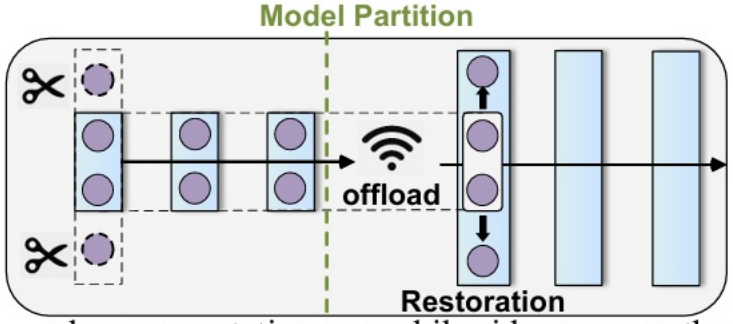
Impact of pruning positions

Mercury: A Framework Towards Optimal Accuracy-Latency Trade-off

- **Sufficient pruning at initial layer** to minimize mobile-side computation and communication latency
- **Reconstruct tokens at the cloud-side** to enhance the inference accuracy



(a) Existing methods: layers on mobile side have heavier computation

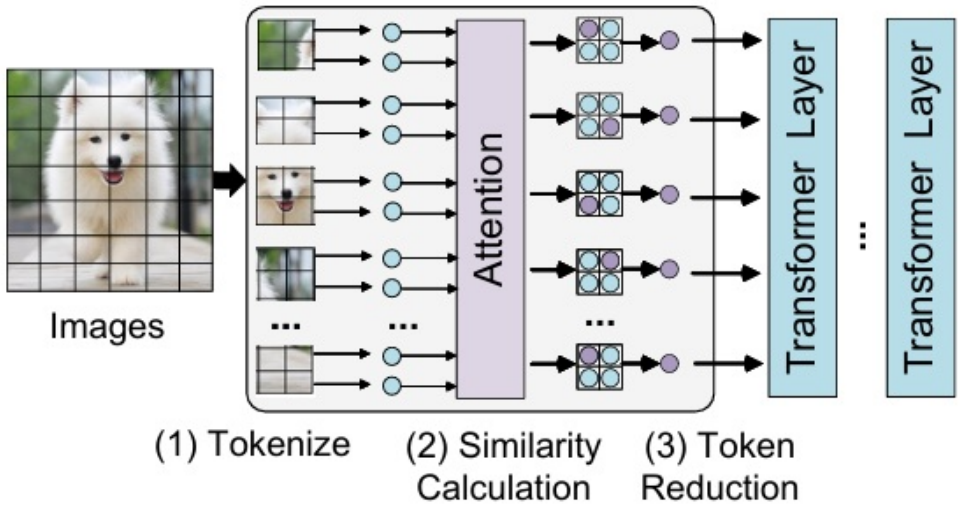


(b) *Mercury*: less computation on mobile side, more on the cloud side

Design1: Similarity-based Regionally Decentralized Token Pruning

Motivation: at initial layer, visual elements are **uniformly scattered** across various subregions

- Regionally Decentralized Pruning: **preserve at least one token** for each subregion



Algorithm 1: Regionally Decentralized Token Pruning

Input: Tokenized features $X = \{x_1, x_2, \ldots, x_N\}$,
Global attention values $O_f = \{o_1, o_2, \ldots, o_N\}$.
Output: Tokens to prune $X_r = \{x_1, x_2, \ldots, x_K\}$.

for $i = 1$ **to** N **do**
  for $j = 1$ **to** N **do**
    Compute the cosine similarity via Eq. 11;
  Compute semantic score for each token via Eq. 12;
for $u = 1$ **to** $\sqrt{N}$ **do**
  for $v = 1$ **to** $\sqrt{N}$ **do**
    Perform 2D gridding via Eq. 13;
for $p = 0$ **to** $\frac{\sqrt{N}}{l} - 1$ **do**
  for $q = 0$ **to** $\frac{\sqrt{N}}{l} - 1$ **do**
    Partition into subregions via Eq. 14;
    Find token with maximum score via Eq. 15;
    Calculate the global index as Eq. 16;
Determine the preserved tokens X^* as Eq. 17;
Determine the pruned tokens X_r as Eq. 18;
return X_r

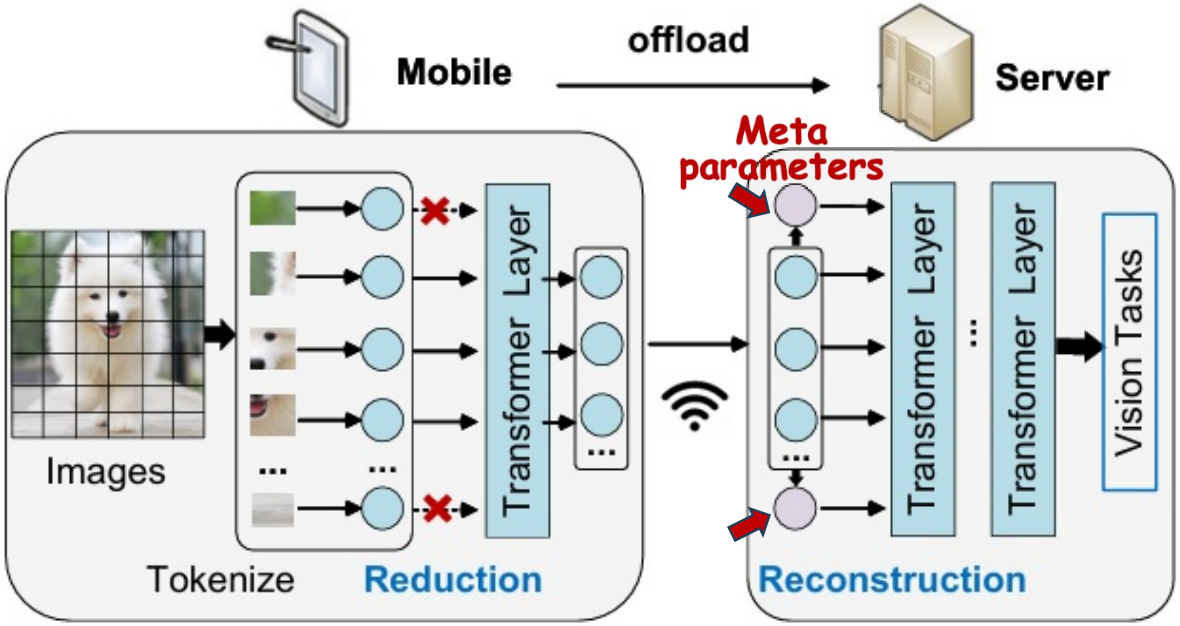
- **Benefits:** the accuracy improves **10.4%~23.1%** under the same pruning rate (50%~75%) compared to existing token pruning methods

Design2: Meta-Parameter based Token Reconstruction

Motivation: Note only missing tokens, but also the reduced computation $O(N^2)$ affects accuracy

- E.g., zero-padding the pruned tokens ($\frac{N}{4}$) to original size (N), improves accuracy by 7.3%

- Using Well-trained Meta-parameters (M) combined with received tokens (X') as input
- The cloud-side computation complex is same as original unpruned one $O(N^2)$,
  without introducing extra computation for generation of missing token.



- **Advantages:** the meta-parameter is less than 0.11M, which is only 0.13% of the original ViT model

Experimental Settings

Mobile device: a Raspberry Pi 4B with 8GB RAM

Cloud device: a server with an NVIDIA 4090 GPU

Real-world Networks Traces: 4G and 5G wireless networks uplink traces[1]

- 126,000 seconds of 5G traces (mean bandwidth 48.2 Mbps)
- 74,000 seconds of 4G traces (mean bandwidth 37.3 Mbps)

ViT models: DeiT-Base (86 M), DeiT-Small (22 M), DeiT-Tiny (5 M)

Task and Dataset: Image classification, CIFAR-100 and ImageNet-1K

Evaluation Metrics: Accuracy, Latency (Computation and Communication)

[1] M. Ghoshal et al, ""An in-depth study of uplink performance of 5g mmwave networks," SIGCOMM Workshop 2022

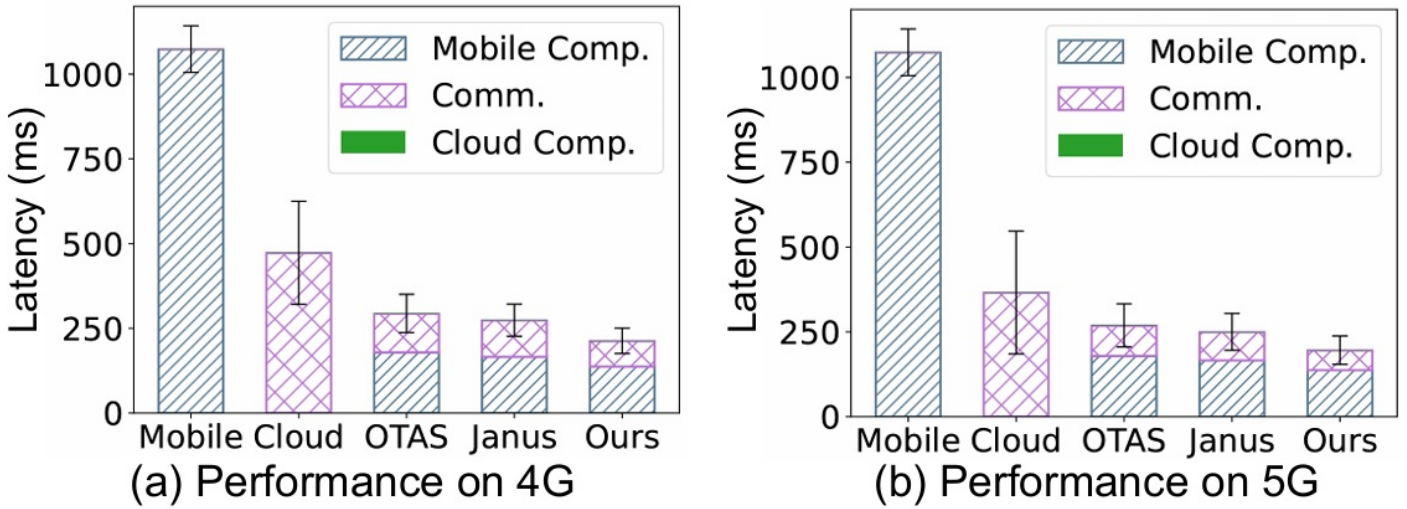
Experimental Results (ViT-Base on ImageNet)

Baselines: **Mobile-only: deploy the entire model on Raspberry Pi**

Cloud-only: transmit the high-quality image to cloud server and processing

OTAS [INFOCOM'24]: elastic adjust Token numbers (reducing or increasing)

Janus [INFOCOM'25]: reduce more tokens on resource-constrained mobile device



Our methods achieves 5.06x and 2.23x speed-up compared to mobile and cloud-only methods

[1] L. Jiang et al, "Janus: Collaborative vision transformer under dynamic network environment," INFOCOM 2025

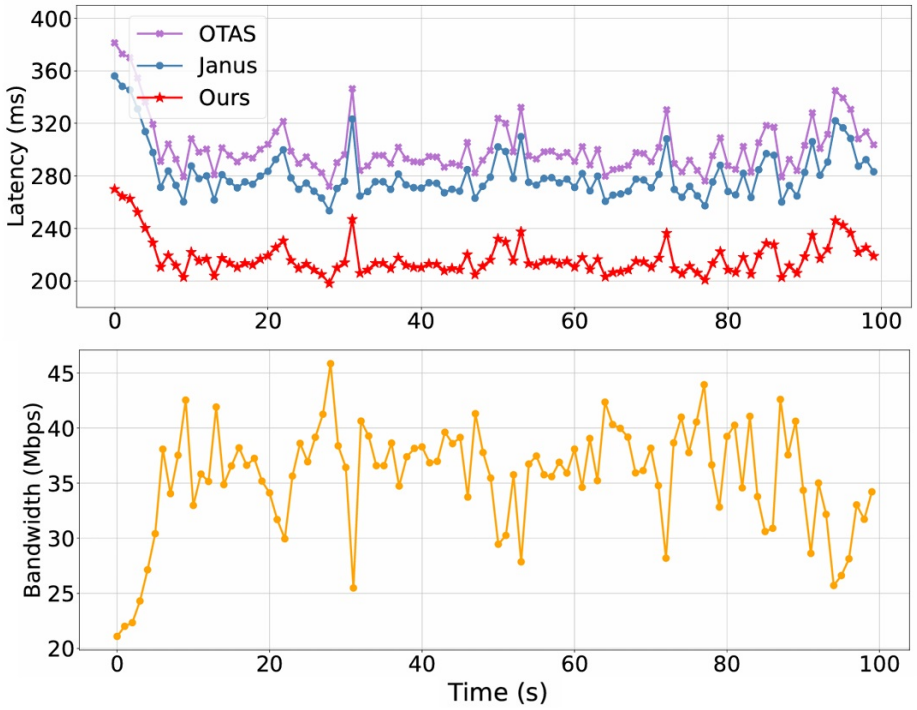
[2] J. Chen et al, "Otas: An elastic transformer serving system via token adaptation," INFOCOM 2024

Experimental Results (ViT-Base on ImageNet)

State-of-the-art: OTAS [INFOCOM'24]: elastic adjust Token numbers (reducing or increasing)

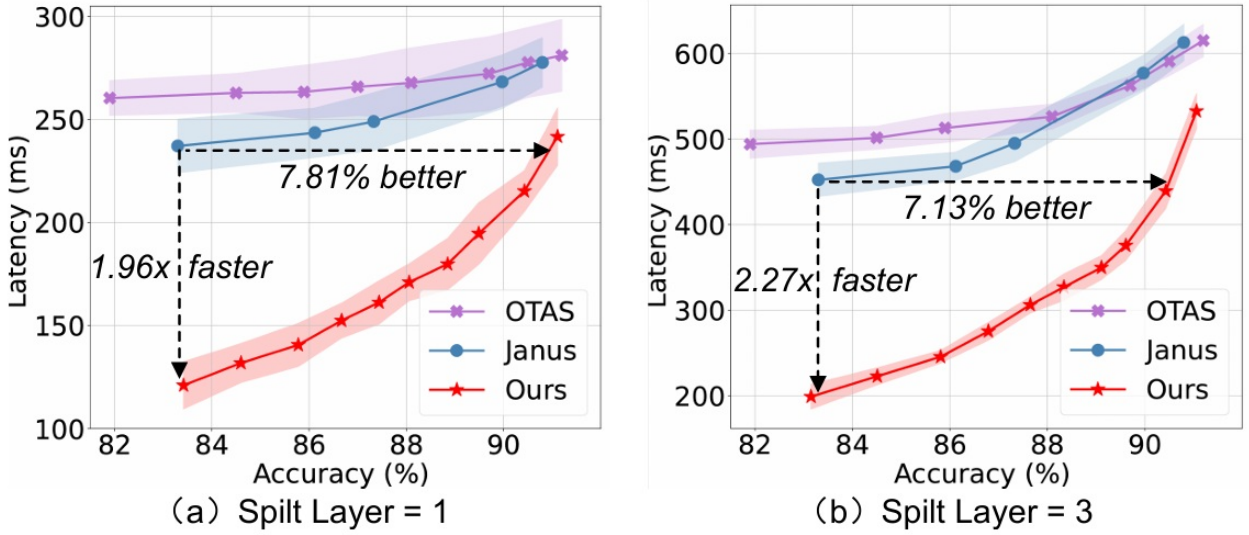
Janus [INFOCOM'25]: reduce more tokens on resource-constrained mobile device

Lower latency under various bandwidth



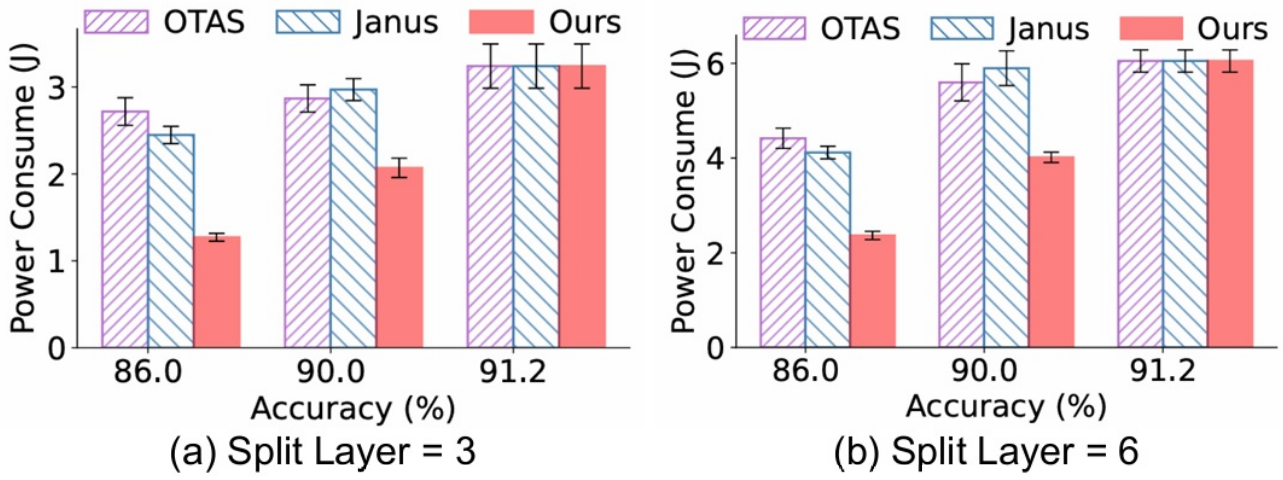
Performance under different bandwidth

Up to 2.27x speed-up under the same accuracy



Accuracy-Latency trade-off

Comparison of Energy Consumption and Overhead



Power consumption of Raspberry Pi

Mobile power consumption is only half of Janus and OTAS, 40% of vanilla model under the same accuracy.

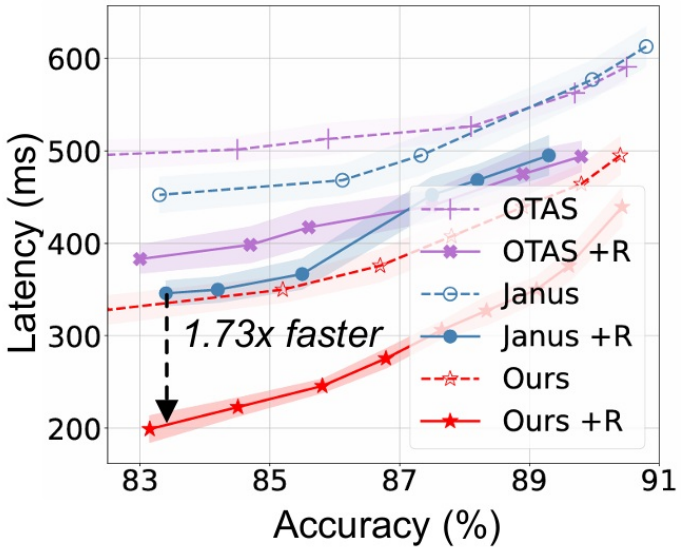
Comparison of Parameters and FLOPS overhead

Method	Accuracy On CIFAR-100	Latency		Parameters on cloud	FLOPs on cloud
		Overall	Cloud-side		
Vanilla	91.2%	636 ms	0.86 ms	63.79 M	26.3 G
Janus [9]	89.9%	577 ms	0.85 ms	63.79 M	20.0 G
Ours	**89.9%**	**407 ms**	**0.86 ms**	**63.84 M**	**26.3 G**

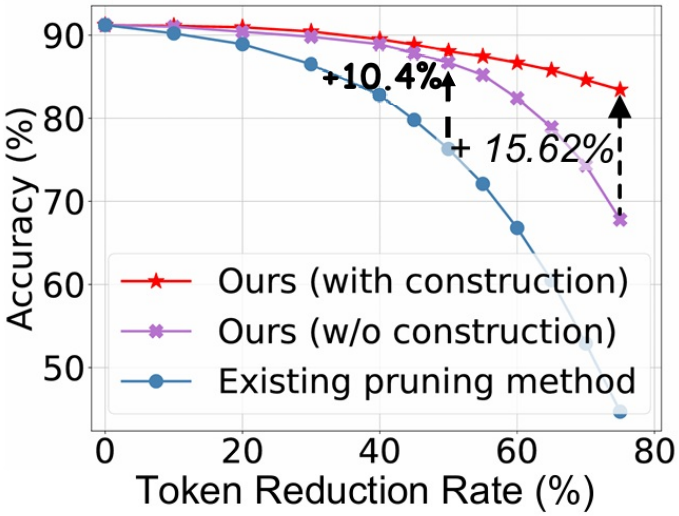
Cloud-side parameter increment is only 0.05M, which is less than 0.1% of the original cloud model

Identical FLOPS with vanilla model

Ablation Study



Impact of Reconstruction(+R)



Impact of pruning method

Only using our pruning and token reconstruction method separately leads to 10.4% higher accuracy, and 1.73x speed-up.

Comparison of Parameters and FLOPS overhead

Method	Accuracy	Parameters	FLOPs
Duplicate	57.71%	63.79 (+0.00) M	26.3 (+0.0) G
Generation	82.55%	73.23 (+9.44) M	28.1 (+1.8) G
Ours	**83.15%**	**63.90 (+0.11) M**	**26.3 (+0.0) G**

25% higher accuracy compared to non-parameter reconstruction (duplicate remaining tokens)

86x fewer parameters increasement compared to generation method

Contribution and Future Work

Contribution:

- Mercury, a novel collaborative inference framework that sufficiently prune tokens on mobile devices and recover them in the cloud to achieve optimal accuracy-latency trade-off, up to 2.27x speedup and 50% mobile energy saving is achieved under same accuracy threshold;
- Similarity-based regionally decentralized pruning strategy, leads to 10.4% higher accuracy;
- Meta-parameter-driven token reconstruction method, leads to additional 15.62% accuracy improvement or 1.73x speed-up.

Future Work:

- Extend to more tasks, e.g., information dense tasks (object detection, segmentation);
- Extend to more ViT models, e.g. Swin Transformer.

中国电信云计算研究院
China Telecom Cloud Computing Research Institute

Thanks

for your listening!