The University of Texas at Austin
Chandra Department of Electrical and Computer Engineering
Cockrell School of Engineering

Energy Aware Computing Research Group

Efficient Learning for Vision Transformers

Diana Marculescu

The University of Texas at Austin

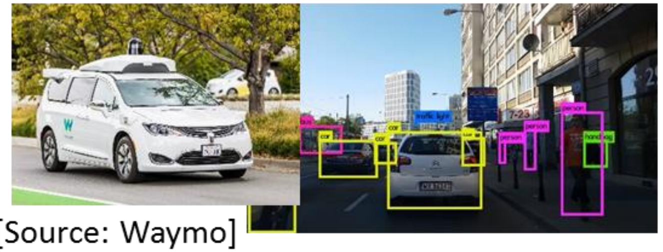
dianam@utexas.edu

enyac.org

Machine learning applications push hardware to its limits

- **ML models are now used in every modern computing system**

Automotive



[Source: Waymo]

[Source: Qualcomm]

Drones

[Source: CNN]

IoT – Smart Home & Security

[Source: Nest]

Mobile & Wearables

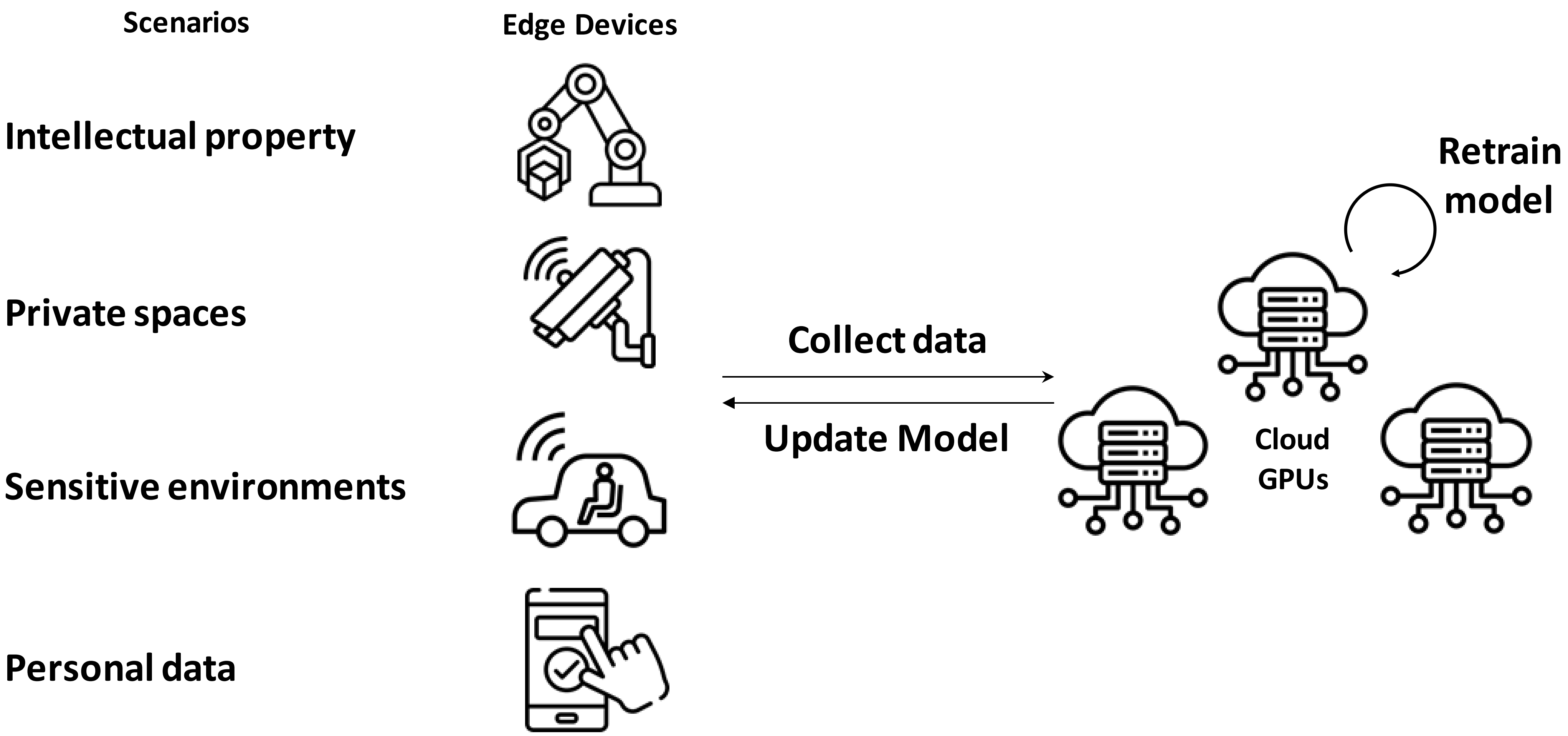
[Source: Qualcomm]

VR/AR

[Source: Asus]

- **Hardware constraints are a key limiting factor for ML on mobile platforms**
  - **Energy** constraints: object detection drains smartphone battery in 1 hour! [Yang *et al., CVPR'*17]
  - Edge-cloud **communication** constraints
  - **On-device inference** (**response**) time constraints AND **expensive on-device training**

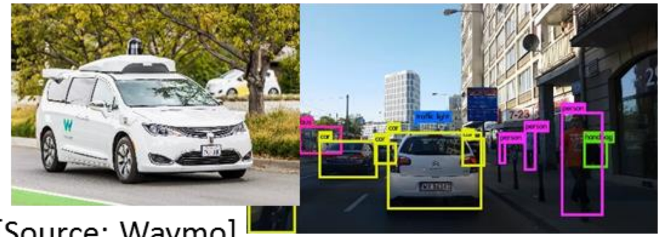
The cloud to edge continuum vs. privacy trade-offs



What about on-device learning?

- **Recall:**

Automotive



[Source: Waymo]

[Source: Qualcomm]

Drones

[Source: CNN]

IoT – Smart Home & Security

[Source: Nest]

Mobile & Wearables

[Source: Qualcomm]

VR/AR

[Source: Asus]

- **Hardware constraints are the key limiting factor for DL on mobile platforms**
  - **Energy** constraints: object detection drains smartphone battery in 1 hour! [Yang *et al., CVPR'*17]
  - Even more **expensive** to do **on-device training**
- **Solution: Transfer learning → adapt the model to the edge device**

Transfer learning on edge is challenging – even for ConvNets

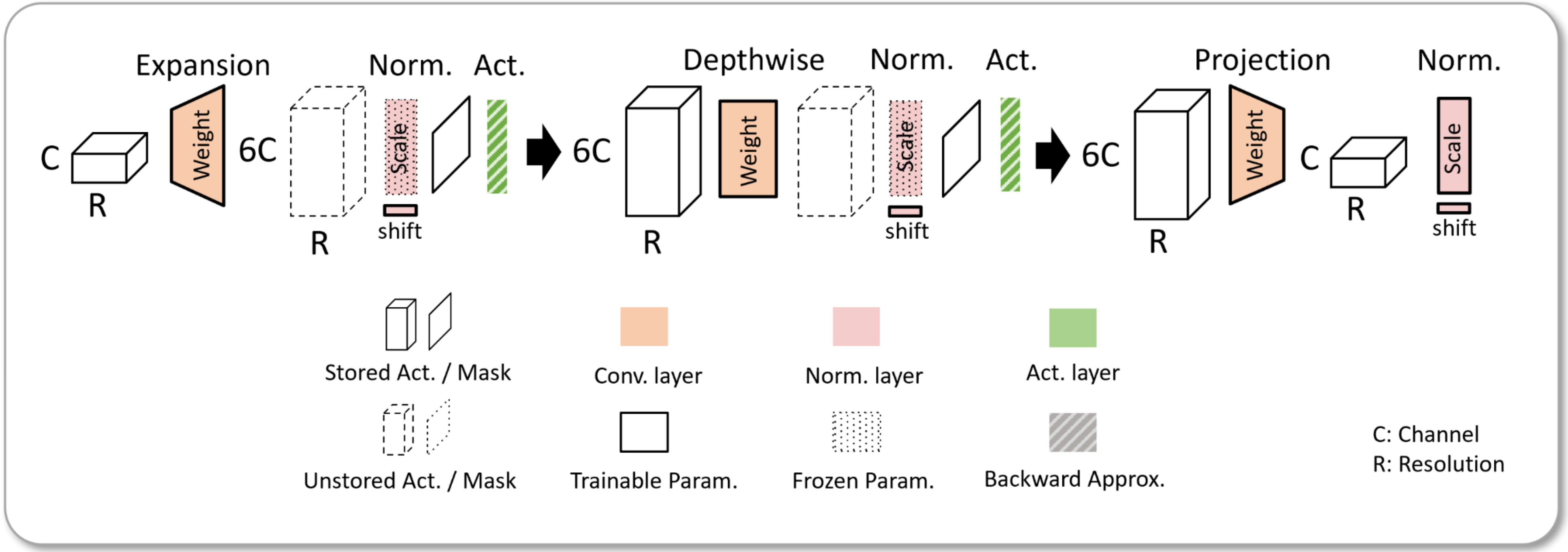
- **Fine-tuning is expensive for large models**
  - Requires careful selection of what is fine-tuned and when

- **Inverted Residual Block (IRB) based models are prevalent on edge**
  - But they require quite a bit of the model resident in memory plus lost of computation

- **Techniques used so far**
  - Freeze certain blocks/layers when fine-tuning
  - Identify which layers are most important for accuracy yet least expensive to fine-tune
  - Are challenging to use under limited hardware constraints

MobileTL: Efficient learning with IRBs

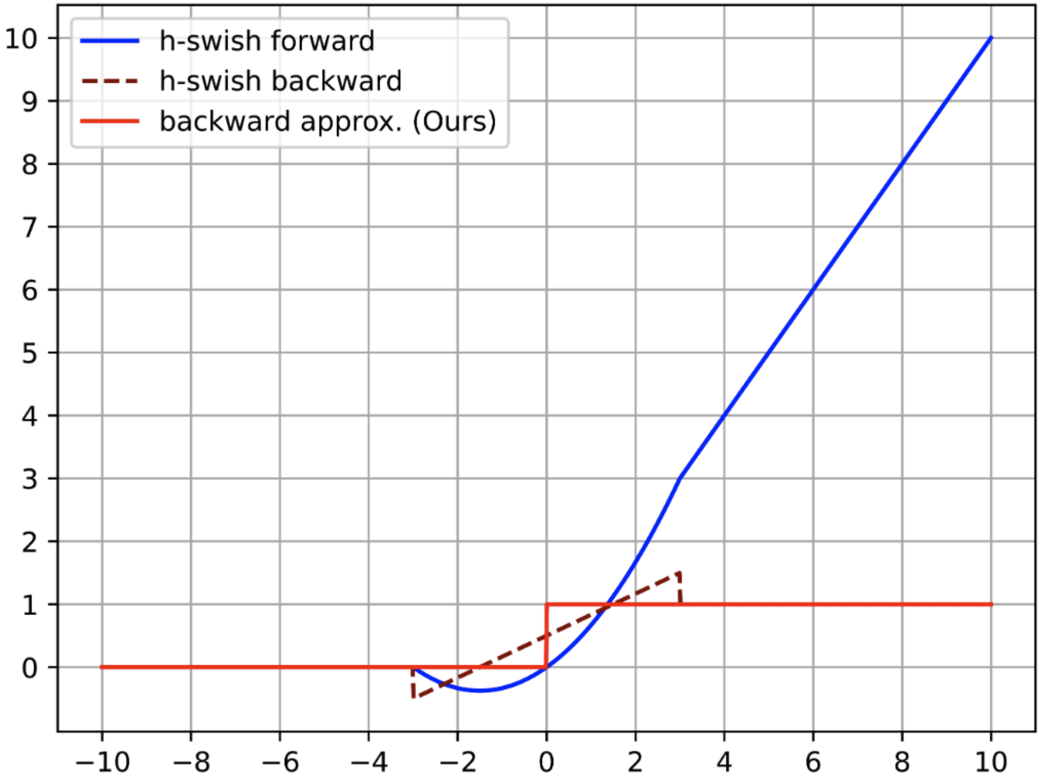
- **Update bias only for intermediate normalization layers**
  - Adapt distribution difference efficiently
- **Approximate activation layer backward as a signed function**
  - Store binary masks for activation layers



[H.-Y. Chiang, N. Frumkin, F. (J.) Liang, D. Marculescu, *AAAI'23*]

Backward activation approximation

- **Backward approximation for Hard-swish activation function**



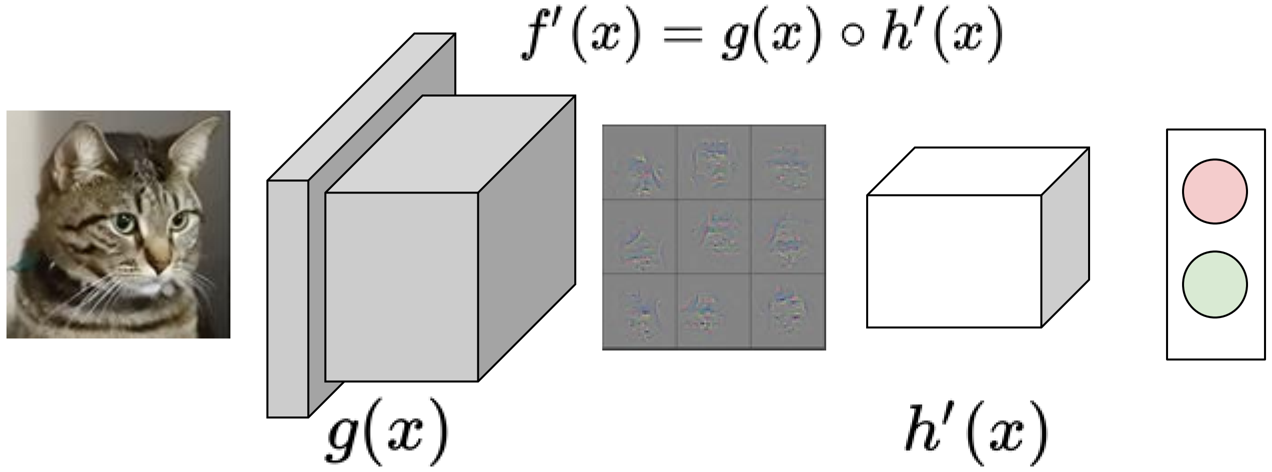
Forward $a_{i+1} = a_i \circ \frac{ReLU6(a_i+3)}{6}$

Backward $\frac{\partial L}{\partial a_i} = \frac{\partial L}{\partial a_{i+1}} \circ \left(\frac{ReLU6(a_i+3)}{6} + a_i \circ \frac{1_{-3 \leq a_i \leq 3}}{6}\right)$

$\boxed{\text{Backward Approx.} \quad \frac{\partial L}{\partial a_i} = \frac{\partial L}{\partial a_{i+1}} \circ 1_{a_i \geq 0}}$ ➡ Store Bitmask

Fine-tune only task-specific blocks

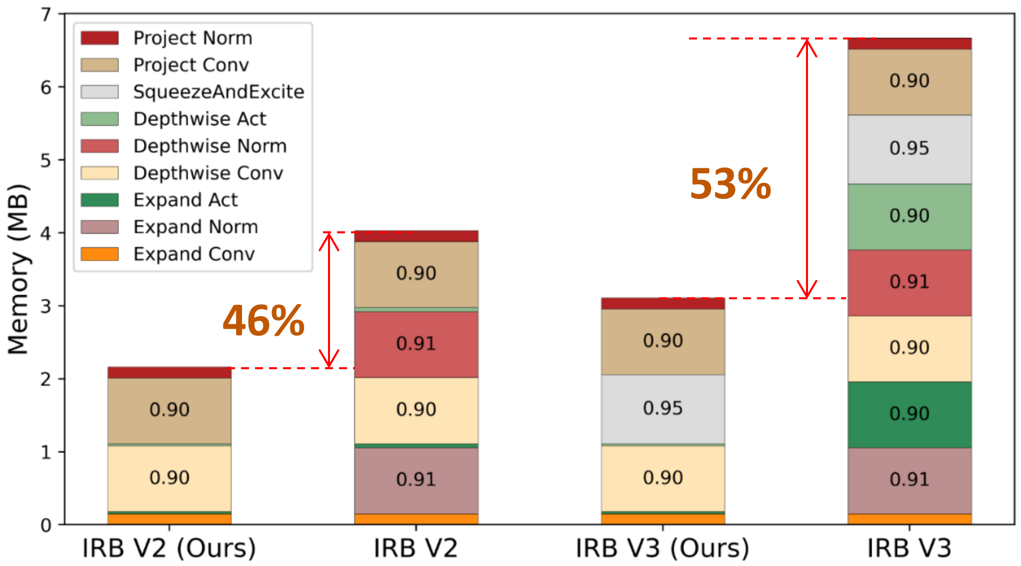
- **Freezes input layers**
  - Low-level features can be shared across different datasets
  - Reduce memory footprint by 8-bit quantization
  - Reduce FLOPs by avoiding calculating gradients for the whole network

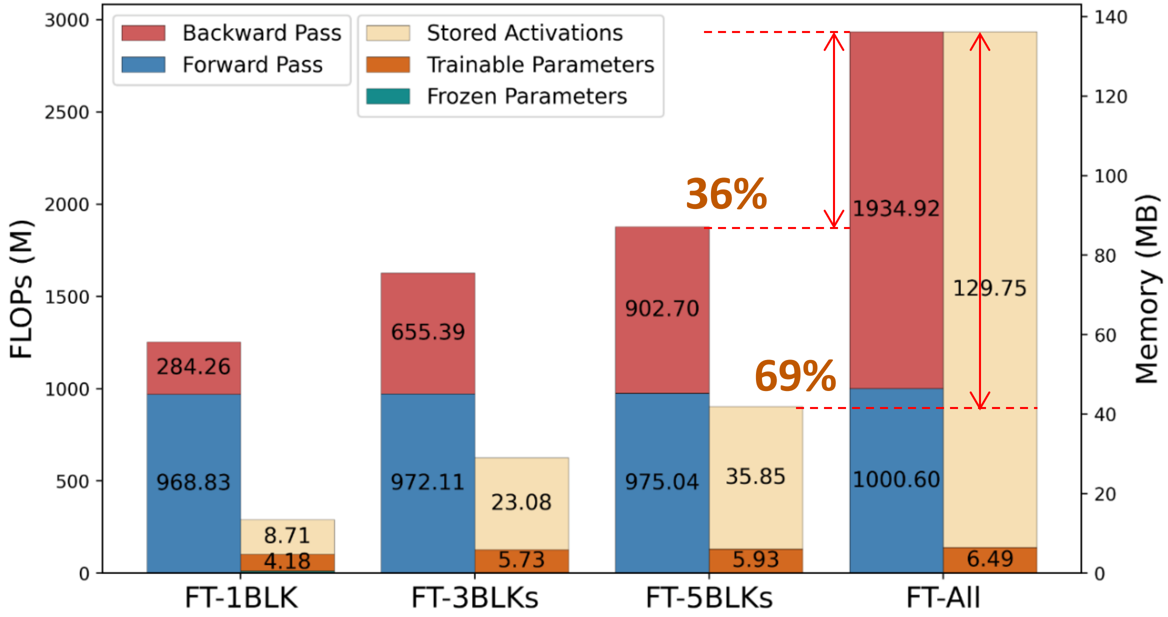


Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." In ECCV, 2014.

Experiments: Less memory and FLOPs

- **Reduce training memory and FLOPs for MobileNetV2 [1] and V3 [2]**

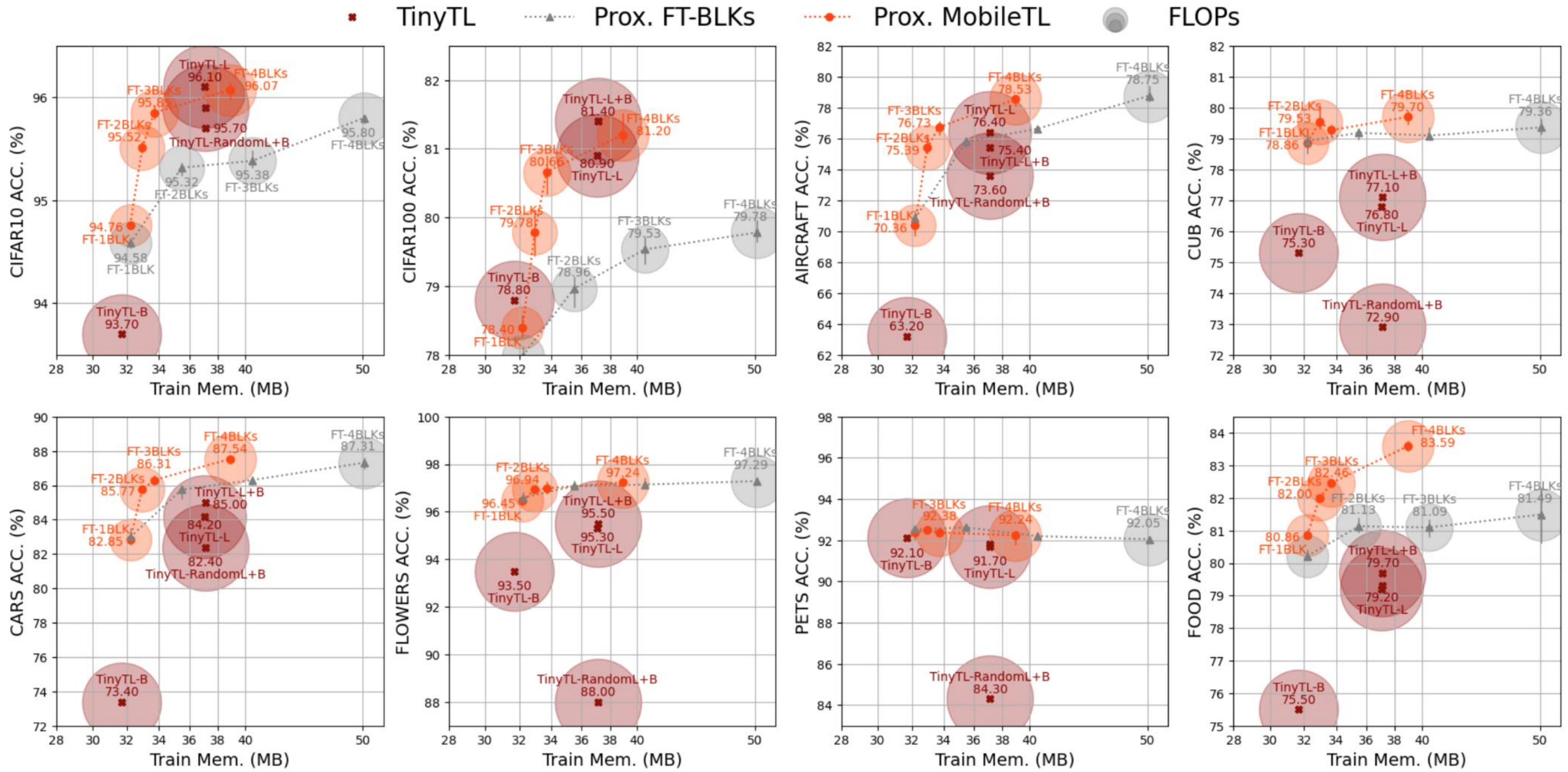




[1] Sandler, M., et al. Mobilenetv2: Inverted residuals and linear bottlenecks. In CVPR, 2018
[2] Howard, A., et al. Searching for mobilenetv3. In ICCV, 2019

Baseline model comparison

- **On the Pareto front under the same memory constraint for various datasets**

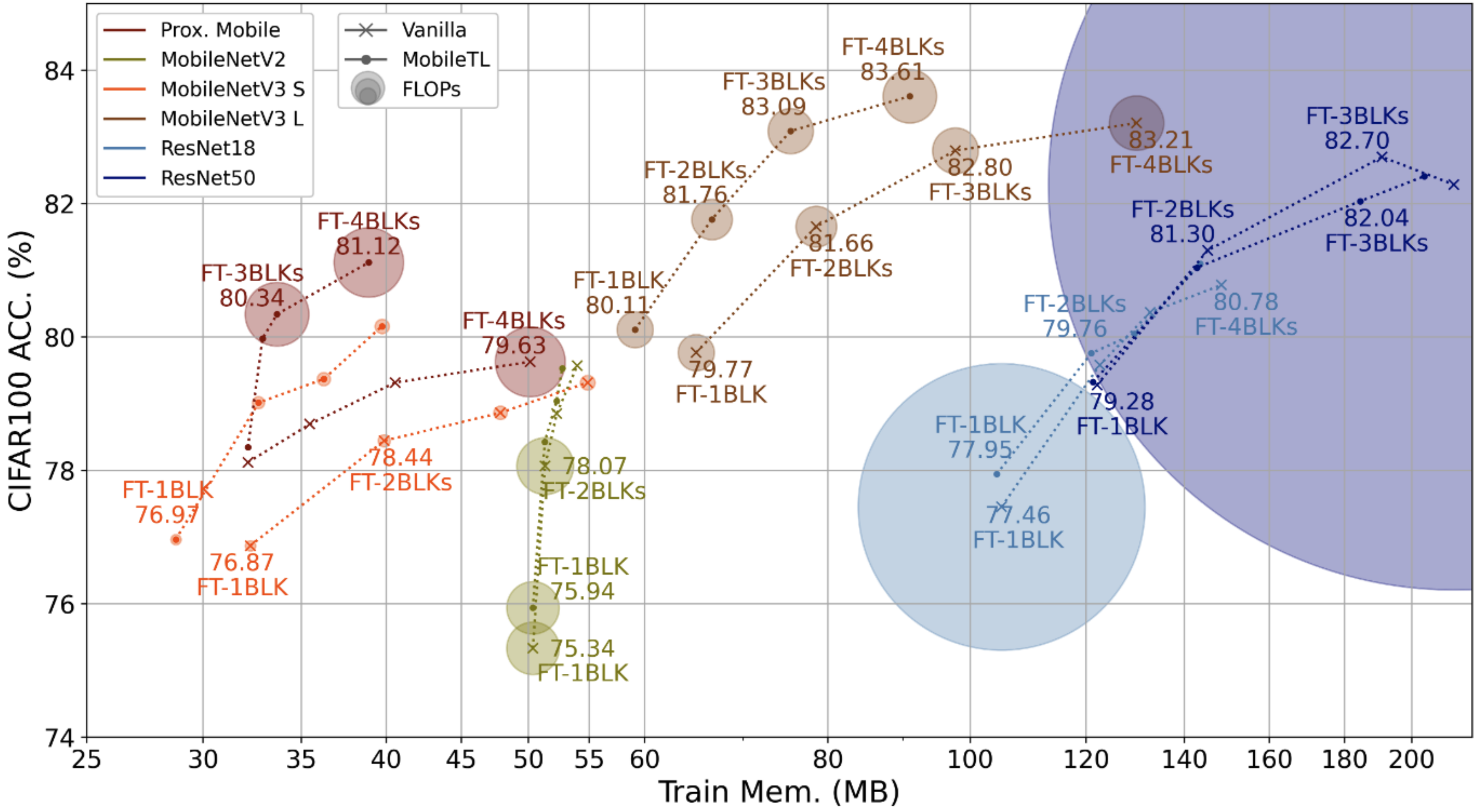


Cai, H., et al. Tinytl: Reduce memory, not parameters for efficient on-device learning. In NeurIPS, 2020
Cai, H., et al. ProxylessNAS: Direct neural architecture search on target task and hardware. In ICLR, 2019

[H.-Y. Chiang, N. Frumkin, F. (J.) Liang, D. Marculescu, *AAAI'23*]

Generalization of MobileTL

- **MobileTL generalizes to off-the-shelf models**



[H.-Y. Chiang, N. Frumkin, F. (J.) Liang, D. Marculescu, *AAAI'23*]

Ablation study

- MobileTL is more effective than patches

Mobile TL	Main Blk	Res. Patch	Train Param.	Mem. (MB)	CIFAR10 (%)
	✓		1,580,682	40.1	95.4
✓	✓		**1,576,074**	**33.2**	**95.8**
✓	✓	✓	2,211,466	35.8	95.8
✓	frozen	✓	1,060,362	32.3	94.4

- MobileTL has lowest latency

Device	Method	Latency (s)
Nano	FT-All	0.235
	FT-BN	0.138
	FT-Bias	0.130
	FT-3BLKs (Ours)	**0.114**
RPI4	FT-All	2.465
	FT-BN	1.894
	FT-Bias	1.818
	FT-3BLKs (Ours)	**1.344**

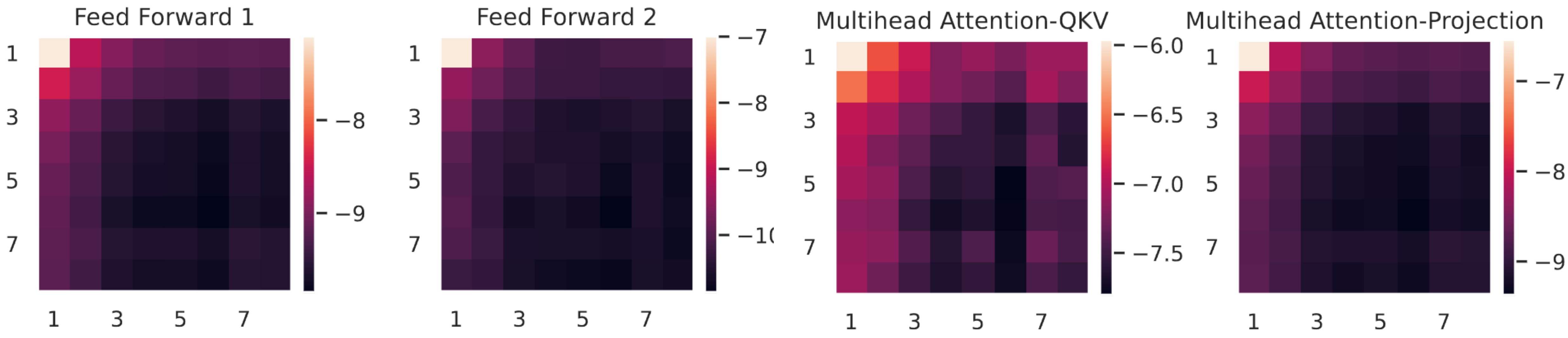
- **45-50% lower latency means *45-50% lower CO_2 footprint***

[H.-Y. Chiang, N. Frumkin, F. (J.) Liang, D. Marculescu, *AAAI'23*]

What about vision transformers (ViTs)?

How can we decrease the computational cost for all operations involved in backpropagation (BP) through any linear layer in the ViT model?

- Accurate Backpropagation is ***NOT*** necessary
- Energy concentrates in low-frequency area (top-left corner)
- Gradient of feature maps can be accurately represented with very few elements in low-frequency area

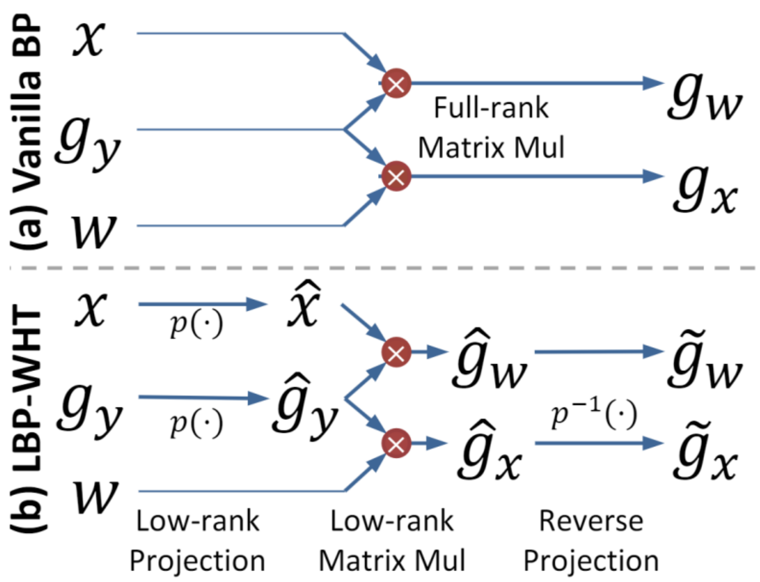


Spectrum of feature gradients in ViT [Unit: db]

LBP-WHT: Low-rank BackProp via Walsh-Hadamard Transformation

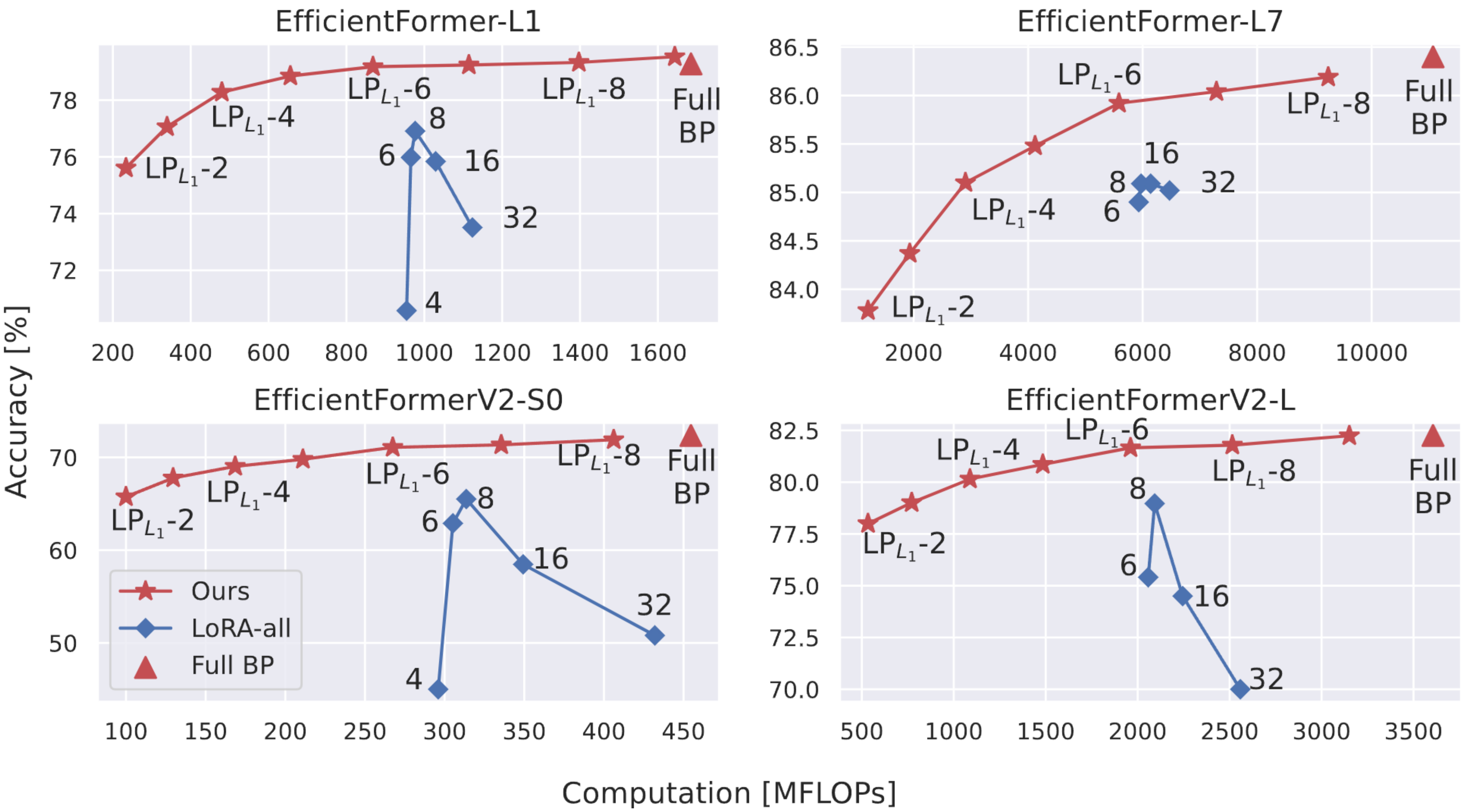
Idea:

- First project gradient into a low-rank space using $p(\cdot)$, then perform matrix multiplications, and finally project them black using $p^{-1}(\cdot)$, where both p and p^{-1} are implemented with WHT



[Y. Yang, H.-Y. Chiang, G. Li, D. Marculescu, R. Marculescu, *NeurIPS'23*]

LBP-WHT is fast and accurate



[Y. Yang, H.-Y. Chiang, G. Li, D. Marculescu, R. Marculescu, *NeurIPS'23*]

LBP-WHT transfers well across multiple tasks

Semantic segmentation on Cityscapes and VOC12 with Segformer

Partial Training: Training Last Stage + Decoder					Full Training				
Method	**R**	**MFLOPs**	**City**	**VOC12A**	**Method**	**R**	**MFLOPs**	**City**	**VOC12A**
Full BP	-	10052.00	62.85	69.30	Full BP	-	16700.26	67.37	70.84
LoRA	8	5854.61	51.43	58.18	LoRA	8	11976.46	62.57	58.18
LoRA-all	8	6262.01	58.07	66.26	LoRA-all	8	11971.13	65.74	67.82
LP_{L_1}-2★	3	**1481.94**	**58.95**	**67.93**	LP_{L_1}-2	3	**5746.54**	61.57	**67.93**
LP_{L_1}-4★	10	**2725.39**	**60.97**	**68.85**	LP_{L_1}-4★	10	**7295.52**	**64.72**	**68.85**
LP_{L_1}-8	36	7308.45	**62.68**	**68.95**	LP_{L_1}-8	36	13086.06	**66.17**	**68.95**

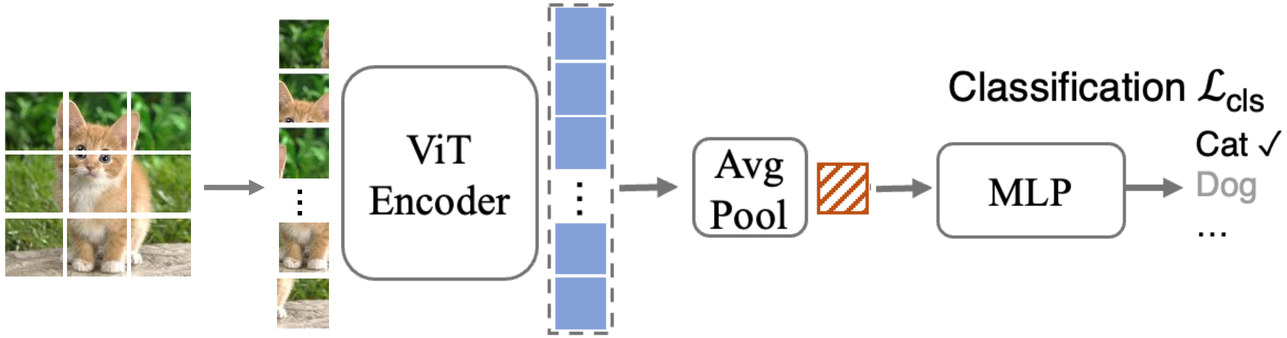
Image classification on CIFAR100 with EfficientFormers

Method	**GFLOPs**	**Memory [MB]**		**Accuracy [%]**	
		Activation	**Gradient**	**CF100**	**CF10**
Full BP	121	141	2352	79.28	95.23
LoRA-all	62	142	44	76.92	94.38
Ours	25	29	2352	78.27	94.60
Ours+LoRA-all	13	29	44	75.48	93.74

[Y. Yang, H.-Y. Chiang, G. Li, D. Marculescu, R. Marculescu, *NeurIPS'23*]

ViTs are hard to train: Can we combine best of both worlds?

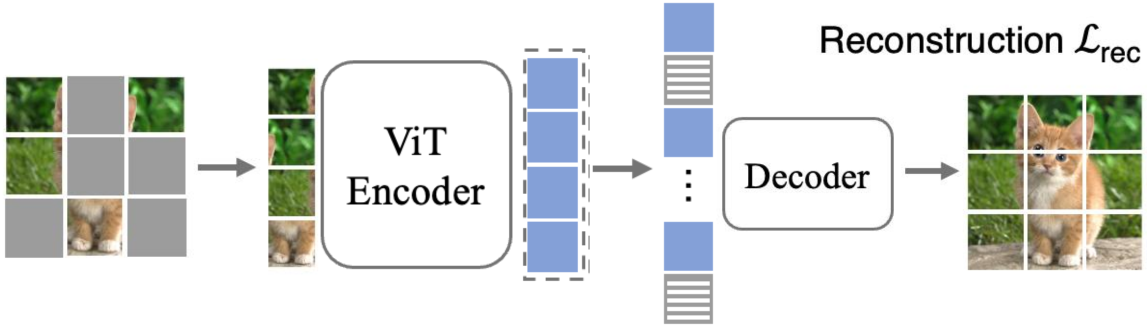
Supervised training



DeiT [H. Touvron *et. al*.]

Training time*	ImageNet acc.
91.5 hours	81.8
✓	✗

Self-supervised pre-training



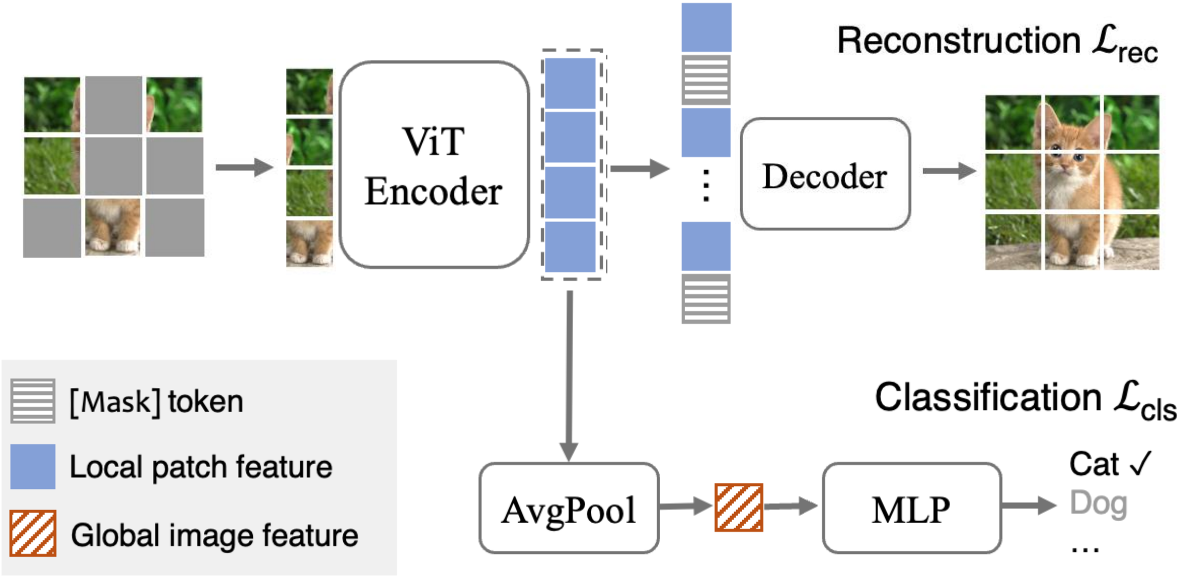
Masked AutoEncoders [K. He *et. al*.]

Training time*	ImageNet acc.+
394 hours	83.6
✗	✓

* Time is measure on 8 A5000 GPUs

+ Accuracy is obtained after supervised fine-tuning on ImageNet

SupMAE achieves the best of both worlds



The proposed SupMAE extends MAE by adding a supervised classification branch

- **Reconstruction loss:** learn middle-level features
- **Classification loss:** learn global features

Training time*	ImageNet acc.+
125.9 hours	83.6
✓	✓

* Time is measure on 8 A5000 GPUs

\+ Accuracy is obtained after supervised fine-tuning on ImageNet

[F. (J.) Liang, Y. Li, D. Marculescu, *EIW-AAAI'24*]

What about model quantization in transformers?

- Quantization enables efficient deployment of models to a variety of inference scenarios

Full Precision Model
(10.2M parameters)

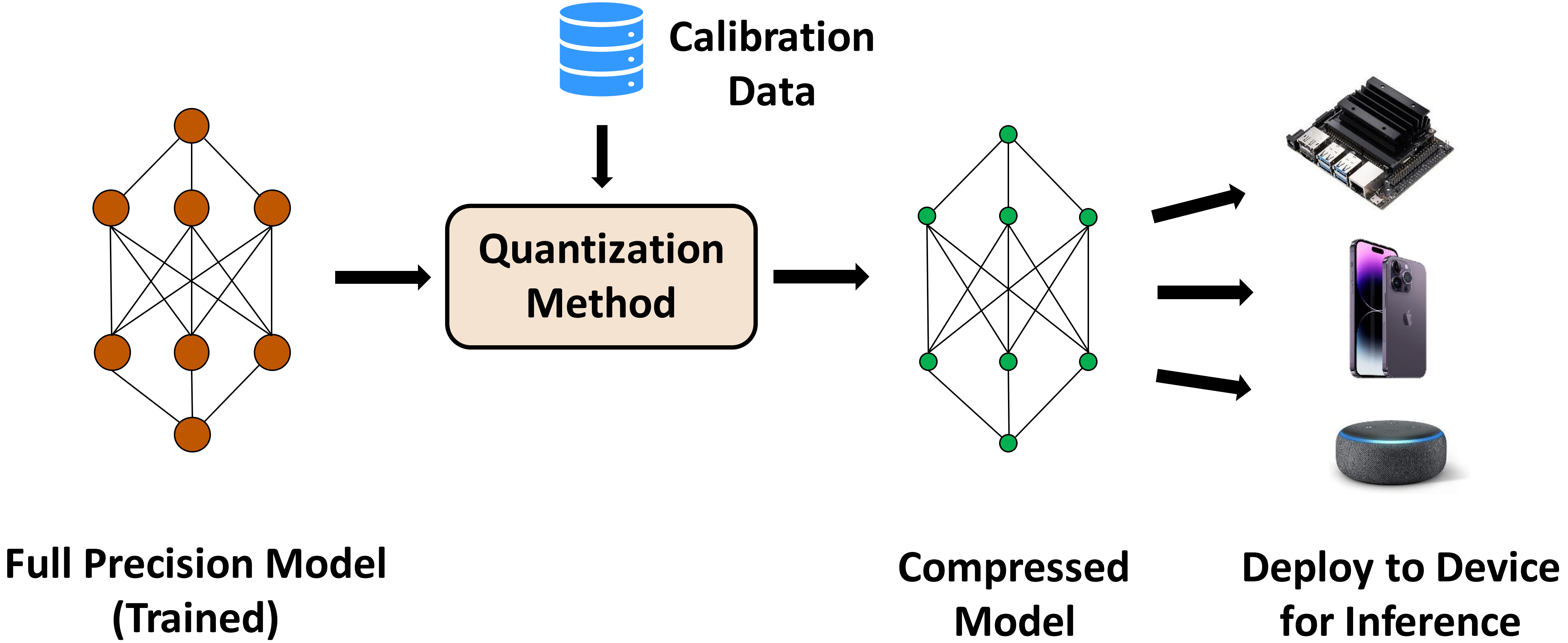
Edge Devices

4-bit Quantized Model
(3.4M parameters)

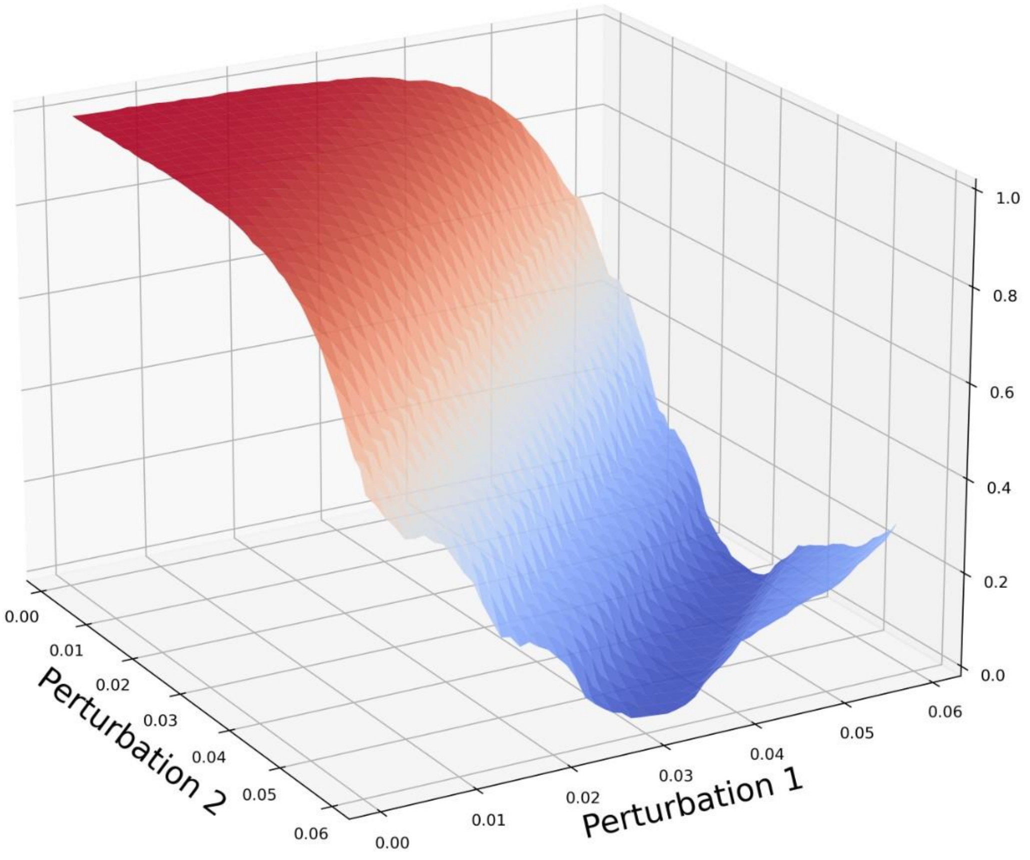
- A compressed model with minimal accuracy degradation is appealing for deployment to edge devices

Post-training quantization (PTQ) for edge deployment

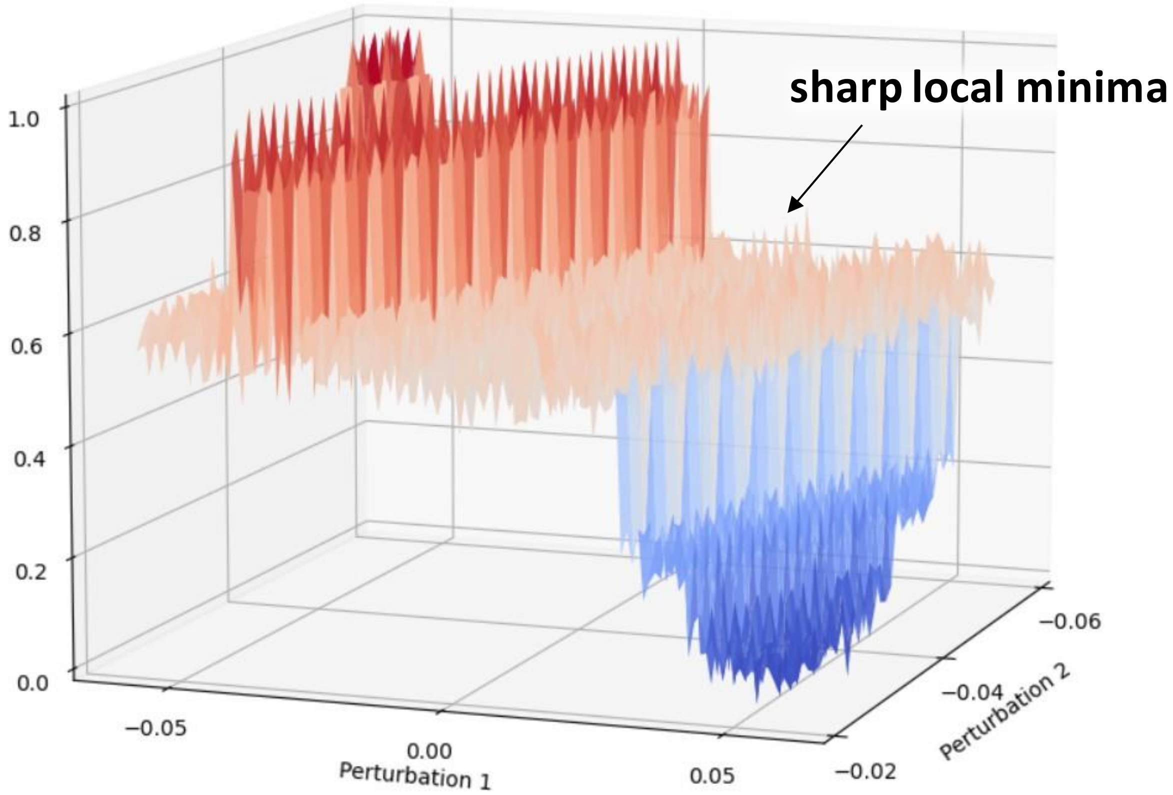
- **The setup for post-training quantization assumes a pre-trained model:**



Quantization in the Loss Landscape of Vision Transformers



Quantized ResNet-18

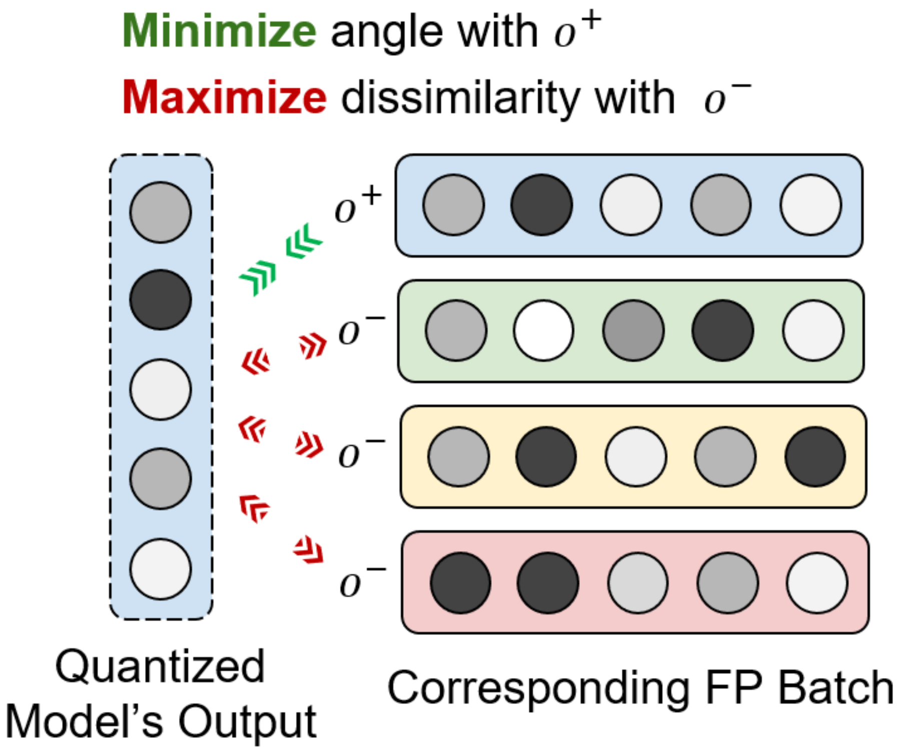


Quantized DeiT-Tiny

[N. Frumkin, D. Gope, D. Marculescu, *ICCV'23*]

Evol-Q: Minimizing a *global objective* using contrastive loss

- Global optimization with a contrastive loss is optimal in our setup



♦ We use the infoNCE loss on network predictions (the final layer's output), and not on intermediary feature maps

[N. Frumkin, D. Gope, D. Marculescu, *ICCV'23*]

Evol-Q: Evolutionary search

- **Recall the uniform quantization formula:**

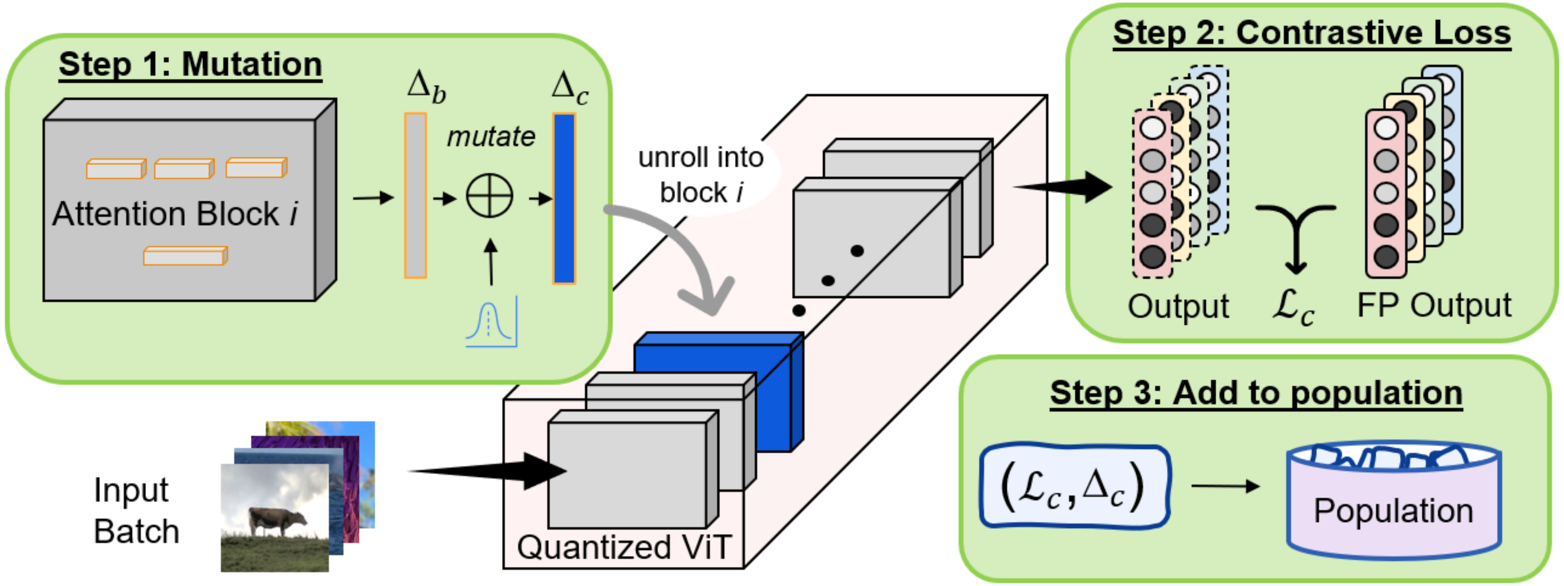
$$Q(\mathbf{x}, \delta, \alpha, \beta) = clip(round(\frac{\mathbf{x}}{\delta}), \alpha, \beta)$$

x	original floating point vector
δ	quantization scale
α, β	quantization range (min, max)

Goal: learn the optimal quantization scales for each attention block

Evol-Q: a fast, effective method for PTQ

- By applying block-wise evolutionary search, we can evaluate small perturbations on quantization scale in a global manner



[N. Frumkin, D. Gope, D. Marculescu, *ICCV'23*]

- Apply block-wise mutation, evaluate using a global contrastive loss

Results on ViTs

- **Top-1 Accuracy on ImageNet for a variety of methods on DeiT and ViT transformers**

8-bit weights, 8-bit activations (8W8A)				
Method	DeiT-T	DeiT-S	DeiT-B	ViT-B
PSAQ-ViT	71.56	76.92	79.10	37.36
PTQ4ViT	-	79.47	81.48	84.25
FQ-ViT	71.61	79.17	81.20	83.31
PSAQ-ViT-V2†	**72.17**	79.56	81.52	-
Evol-Q (ours)	71.63	**79.57**	**82.67**	**84.40**

† Does not quantize Softmax/GELU layers

4-bit weights, 8-bit activations (4W8A)				
Method	DeiT-T	DeiT-S	DeiT-B	ViT-B
PSAQ-ViT	65.57	73.23	77.05	25.34
PTQ4ViT	-	-	64.39	-
FQ-ViT	66.91	76.93	79.99	78.73
PSAQ-ViT-V2†	**68.61**	76.36	79.49	-
Evol-Q (ours)	67.29	**77.06**	**80.15**	**79.50**

† Does not quantize Softmax/GELU layers

- **PSAQ-ViT-V2 achieves comparable accuracy, but is not end-to-end**

[N. Frumkin, D. Gope, D. Marculescu, *ICCV'23*]

Results on ViTs

- **Top-1 Accuracy on ImageNet for LeViT models**

Model	FQ-ViT	Evol-Q (ours)
LeViT-128S	14.90	**29.20**
LeViT-192	17.00	**30.37**
LeViT-256	61.33	**64.57**
LeViT-384	64.60	**69.50**

- **FQ-ViT is effective on standard ViTs, but Evol-Q can bridge the gap to different vision transformer architectures**

[N. Frumkin, D. Gope, D. Marculescu, *ICCV'23*]

Comparison with Gradient Methods

Method	DeiT-T	DeiT-S	DeiT-B	ViT-B
SGD	71.57	79.25	81.24	83.40
Adam	71.29	79.25	81.24	83.25
AdamW	71.37	79.00	81.30	83.36
Evol-Q (ours)	**71.63**	**79.57**	**82.67**	**84.40**

- **Evol-Q improves over gradient-based methods, suggesting that gradient information does not point to a good local minima in the non-smooth loss landscape**

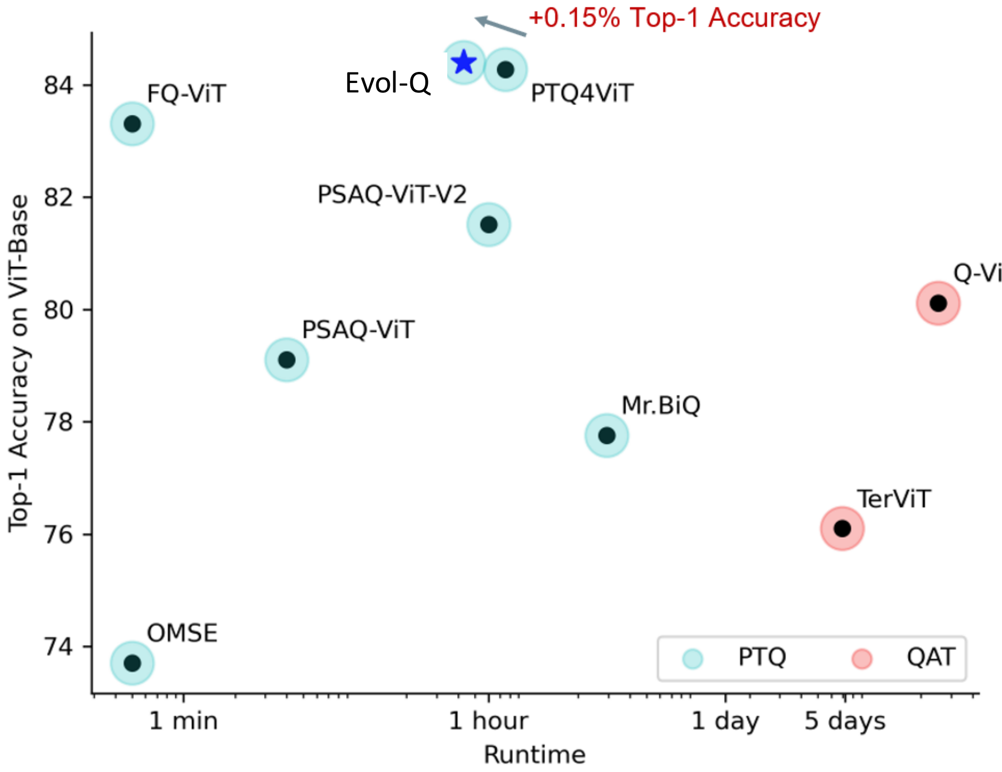
[N. Frumkin, D. Gope, D. Marculescu, *ICCV'23*]

Latency vs. accuracy trade-off

- Evol-Q is pareto-optimal with respect to prior ViT quantization work

Evol-Q's runtime on Nvidia A100

	DeiT-T	DeiT-S	DeiT-B	ViT-B
Runtime (mins)	41.5	46.3	41.6	43.2



[N. Frumkin, D. Gope, D. Marculescu, *ICCV'23*]

Summary

- ViTs can offer higher performance than ConvNet models but at a high computational cost
- MobileTL helps with reducing cost for on-device learning, and similar work for ViTs relying on low-rank backprop like LBP-WHT achieves both accuracy and speed
- Post-training quantization in ViTs with Evol-Q increases efficiency of on-device deployment at no drop in performance

The University of Texas at Austin
Chandra Department of Electrical and Computer Engineering
Cockrell School of Engineering

Energy Aware Computing Research Group

Thank you!

Questions

Acknowledgements:

Students: Hung-Yueh Chiang, Natasha Frumkin, Jeff Liang, Tanvir Mahmud

Support: National Science Foundation, Office of Naval Research (Minerva), iMAGiNE Consortium at the University of Texas at Austin

EnyAC group webpage: enyac.org **Code: github.com/enyac-group**

iMAGiNE
INTELLIGENT MACHINE ENGINEERING CONSORTIUM