IDSL
Intelligent Digital Systems Design Lab.

서울과학기술대학교
SEOULTECH

Performance Optimization for Deep Neural Networks

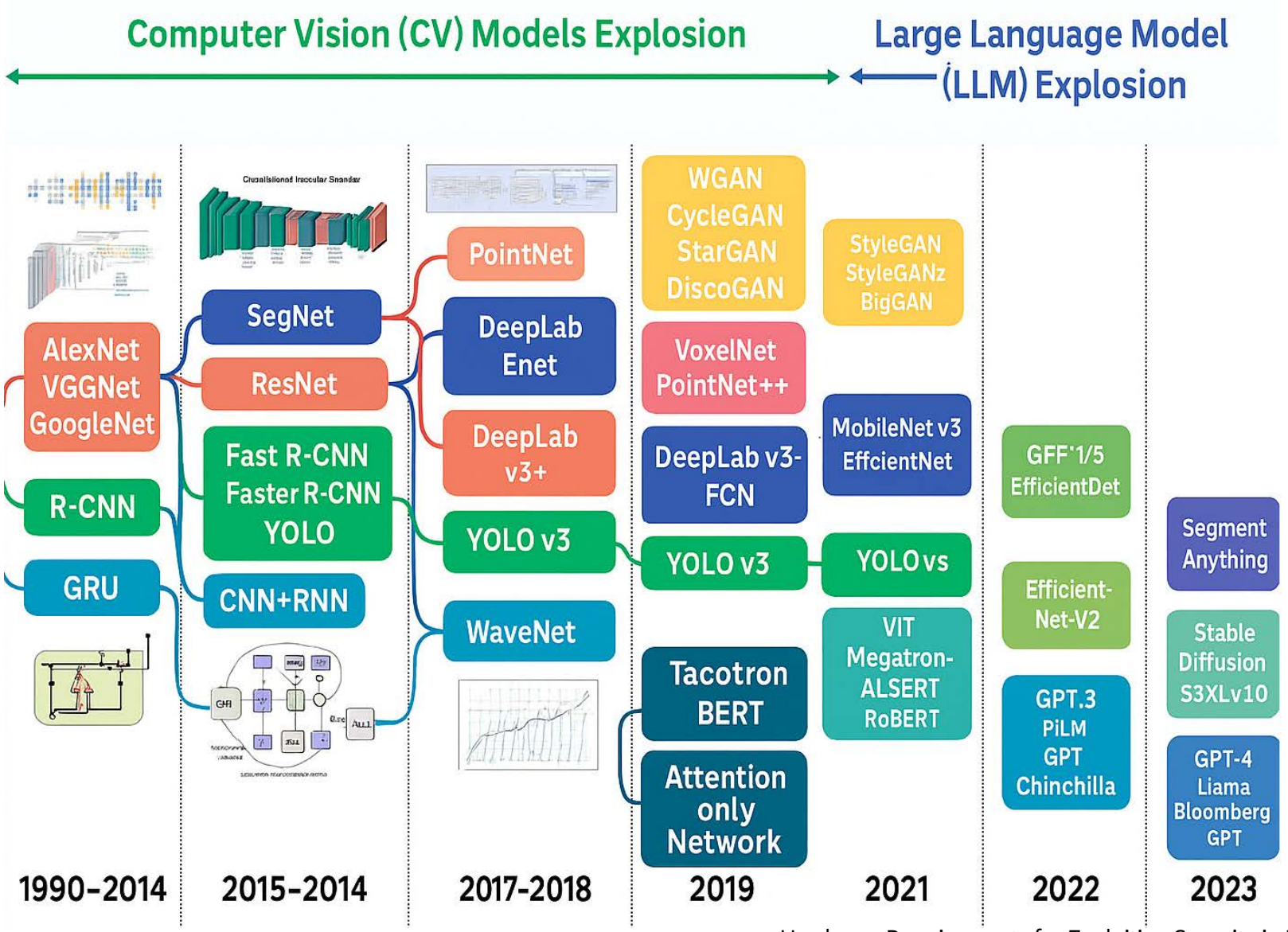
Presenter **Hyun Kim** | **Associate Professor**

Affiliation Seoul National University of Science and Technology
Electrical and Information Engineering

Contact hyunkim@seoultech.ac.kr / idsl.seoultech.ac.kr

A Brief History of AI Models

The paradigm for AI models is shifting from CV to LLM

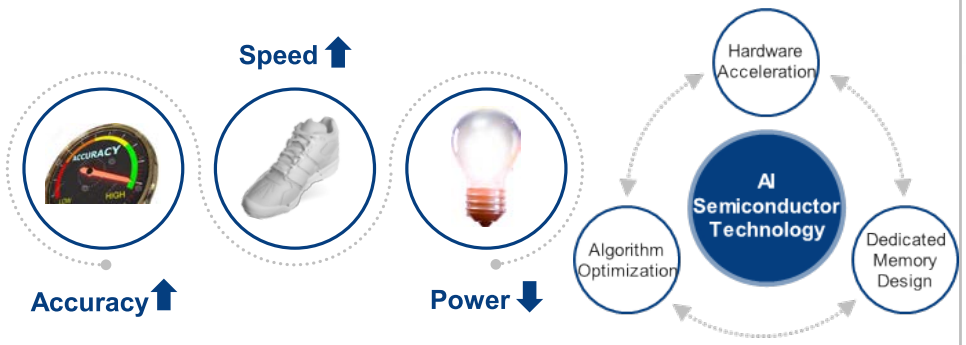


Hardware Requirements for Exploiting Sparsity in ML Inference, Hotchips 2023

6 Key Issue of On-device AI Accelerators: Algorithm

Three main goals of AI accelerators

High accuracy + High speed (Throughput) + Low power (Energy-Efficiency)



Necessity of algorithm-level approach

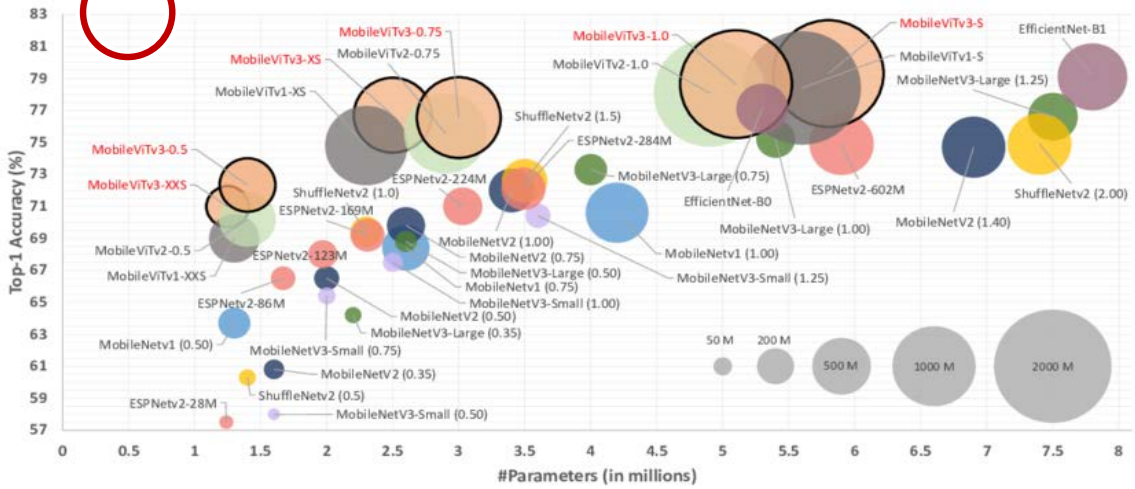
Network optimization is the only way to **improve accuracy** and **reduce the network size** (considering the trade-off between accuracy and computation cost) → Reducing the network size leads to significant power savings and speed-up

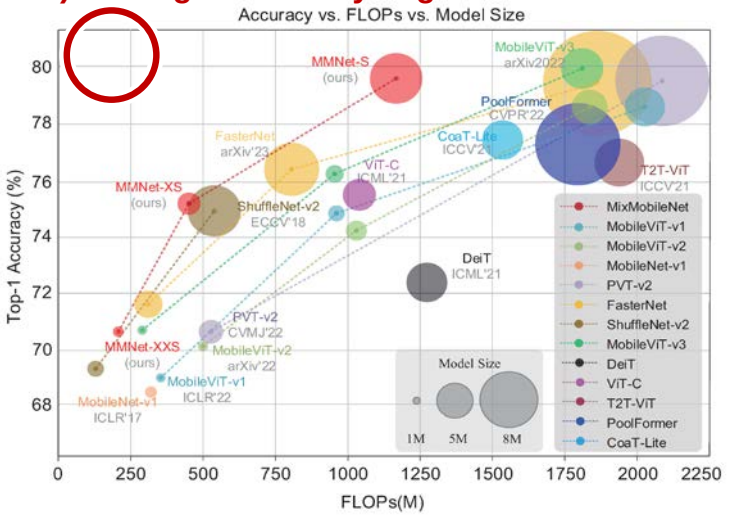
Key Point

HW-friendly model compression + **Performance improvement** w/o increasing model size

Optimal AI models considering the trade-off between accuracy and computation cost/model size

Developing models in this position (High accuracy w/ less computation) is the goal of every engineer!



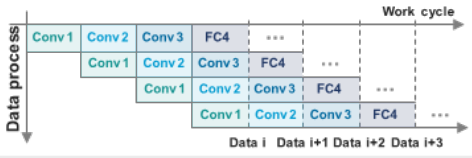


Overview of On-device AI Accelerators

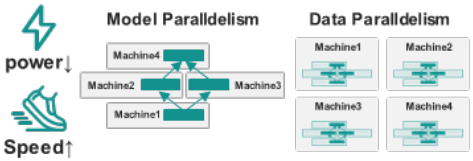
Mobile Characteristic

- Both Inference & Training
- Low-Power FPGA/ASIC for Mobile
- Low Precision: 2b/4b/8b (INT)
- Sparse network
- Application-specific accelerator design

HW-based low complexity schemes for low-power & speed-up

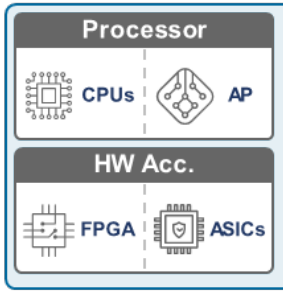


Pipelining

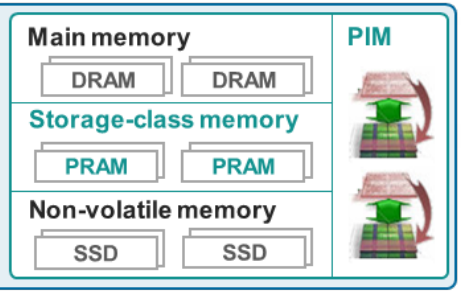


Parallelism

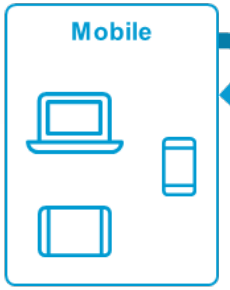
Architecture Platform



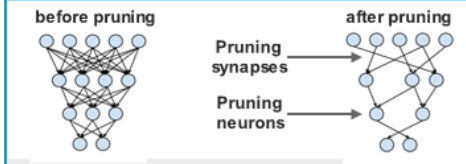
Memory System for DNNs



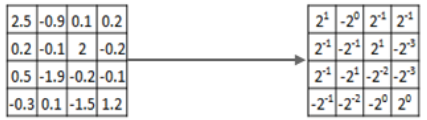
Self-Learning



SW-based low complexity schemes for low-power & speed-up

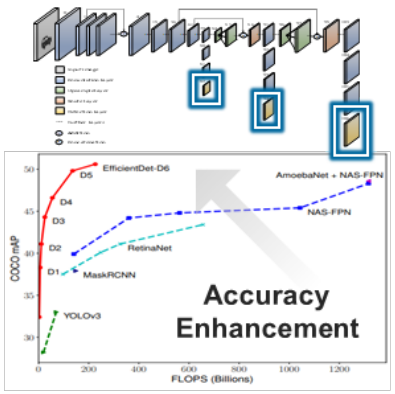


Pruning



Quantization

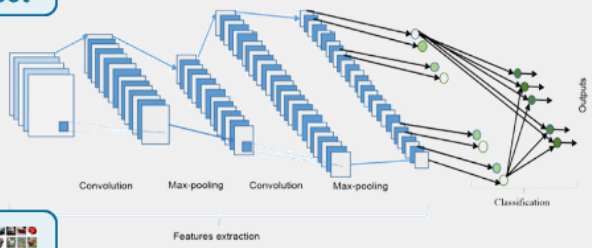
Performance enhancement schemes

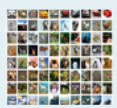


Deep Neural Networks

Test set

Forward (for Inference & Training)





Training set

Backward (for Training)

Apply to target applications

Autonomous Driving

Robot (Physical AI)

Gaussian YOLOv3: Gaussian Modeling

Goal

Improve the detection accuracy with negligible increase in computations

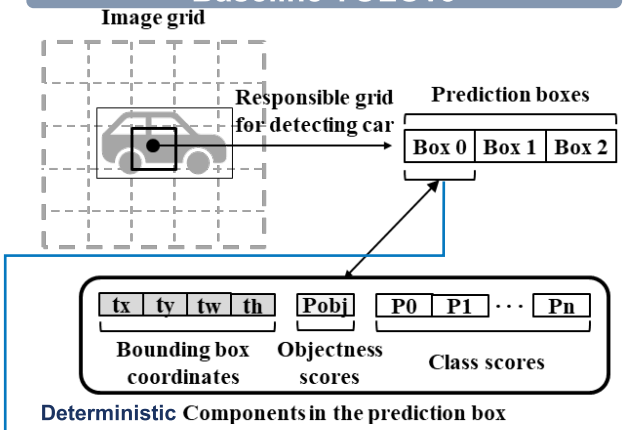
Motivation

- Conventional object detectors output **only the deterministic results of bbox coordinates** → **They cannot determine whether the predicted bbox is correct or not**

Solution/ Contribution

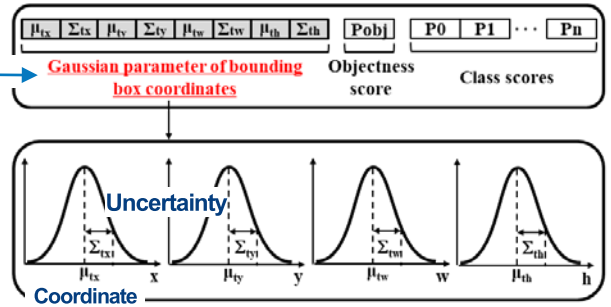
1. **Modeling each bbox coordinate of existing YOLOv3 as Gaussian parameter**
2. **Utilization of uncertainty:** Including (1-Uncertainty) in detection criteria and non-max suppression steps

Components of the Prediction Bbox of Baseline YOLOv3

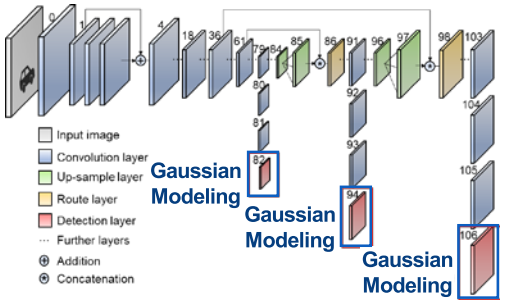


Deterministic Components in the prediction box

Components of the Prediction Bbox of the Proposed Gaussian YOLOv3



Layer Composition of G-YOLOv3



Experimental Results on KITTI validation set (Left) and BDD test set (Right)

Detection Algorithm	mAP (%)	FPS	Input size
MS-CNN	84.7	8.1	1920×576
SINet	85.4	24.0	1920×576
SSD	61.3	28.9	512×512
RefineDet	84.4	27.8	512×512
CFENet	-	0.3	512×512
RFBNet	73.4	39.2	512×512
YOLOv3	80.5	43.6	512×512
Gaussian YOLOv3	**83.6**	**43.1**	**512×512**

3.1% mAP enhancement Negligible increase in computations

Detection Algorithm	mAP (%)	FPS	Input size
MS-CNN	5.7	6.0	1920×576
SINet	9.0	18.2	1920×576
SSD	14.1	23.1	512×512
RefineDet	17.4	22.3	512×512
CFENet	19.1	21.0	512×512
RFBNet	14.5	39.0	512×512
YOLOv3	14.9	42.9	512×512
Gaussian YOLOv3	**18.4**	**42.5**	**512×512**

3.5% mAP enhancement Negligible increase in computations

"Gaussian YOLOv3: An Accurate and Fast Object Detector using Localization Uncertainty for Autonomous Driving," ICCV 2019 (BK21 Top-tier Conf., Citation: 640)

Uncertainty-based Semi-supervised Object Detection

Goal

Improve the pseudo-label reliability with uncertainty

Motivation

Recent DNNs are overconfident → **Generating pseudo-labels with them reduces reliability due to noise**

Solution/ Contribution

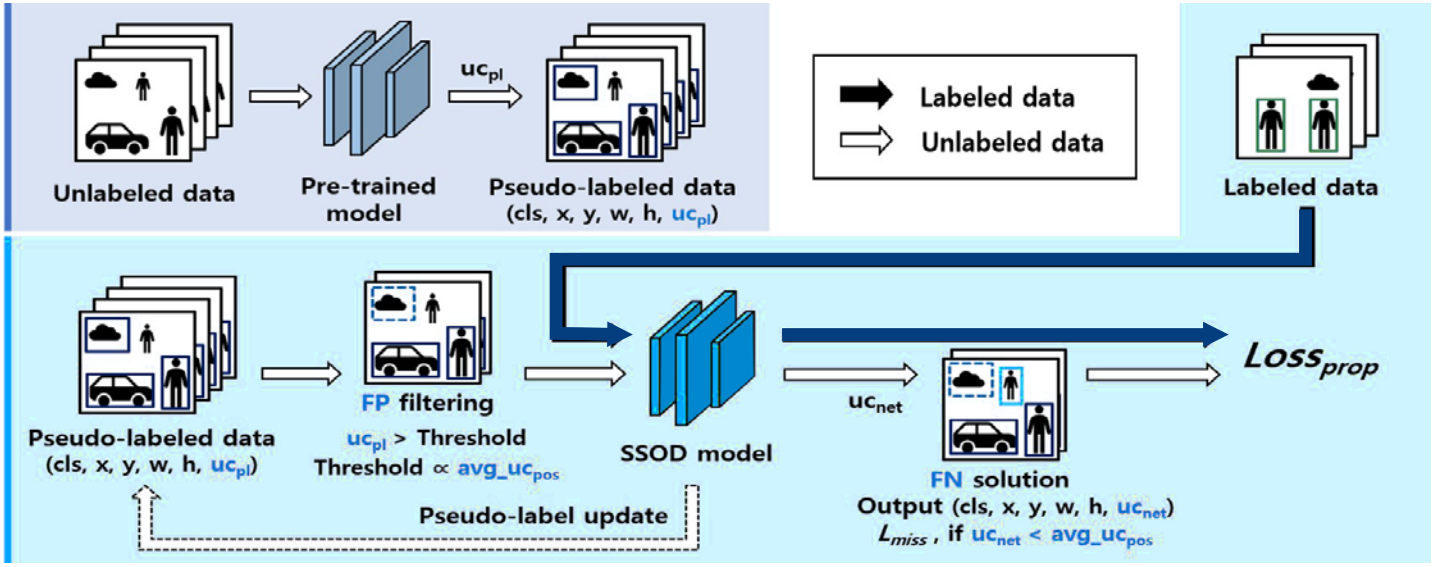
1 False Negative solution

Design uncertainty-weighted loss function & Training missing(FN) objects as positive samples

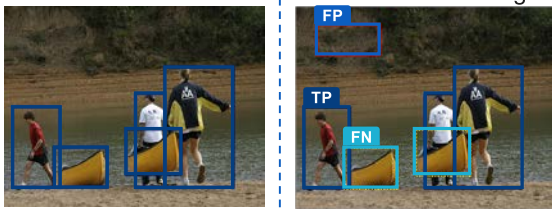
2 False Positive solution

Select highly reliable annotations using uncertainty-based adaptive image-level filtering with threshold (avg_ucpos*: Mean uc_{pos} of predicted positives)

The proposed SSOD training mechanism



Example of miss-annotation



- **TP** True Positive: A box correctly detected
- **FN** False Negative: A box missed
- **FP** False Positive: A box wrongly detected

Detection results on Pascal VOC07

Model	Method	L	Un	mAP(%)	FLOPs(x10^{12})
SSD300 (VGG16)	FS	VOC07	-	70.2	-
	Baseline	VOC07	VOC12	71.8	1.10
	CSD [1]	VOC07	VOC12	72.3	1.56
	ISD [2]	VOC07	VOC12	74.4	2.02
	MP [3]	VOC07	VOC12	72.3	1.71
	MP+ [3]	VOC07	VOC12	74.5	-
	ours	VOC07	VOC12	73.4	1.17
	ours+	VOC07	VOC12	75.1	-
	FS	VOC07	VOC12	77.2	-
Tiny-YOLOv3	FS	VOC07	-	30.67	-
	Baseline	VOC07	VOC12	33.19	0.87
	MP [3]	VOC07	VOC12	33.91	1.32
	MP+ [3]	VOC07	VOC12	37.28	-
	ours	VOC07	VOC12	36.53	0.88
	ours+	VOC07	VOC12	37.29	-
	FS	VOC07	VOC12	71.8	-
Faster R-CNN (ResNet-50)	FS	VOC07	-	74.8	-
	Baseline	VOC07	VOC12	75.6	9.89
	CSD [1]	VOC07	VOC12	74.7	13.94
	ISMT [4]	VOC07	VOC12	77.2	271.15
	UT [5]	VOC07	VOC12	77.4	12.98
	Rethinking [6]	VOC07	VOC12	76.9	9.94
	MP [3]	VOC07	VOC12	77.4	14.83
	MP+ [3]	VOC07	VOC12	78.6	-
	ours	VOC07	VOC12	77.8	9.94
	ours+	VOC07	VOC12	78.7	-
	FS	VOC07	VOC12	81.2	-

Negligible increase in computations

"USD: Uncertainty-based One-phase Learning to Enhance Pseudo Label Reliability for Semi-supervised Object Detection," IEEE TMM, 2024 (IF 9.7)

Adaptive Gradient Quantization for Low-Bit CNN Training

Goal

A CNN gradient quantization framework for on-device training

Motivation

Small and large magnitude gradients are crucial for stable CNN training and stochastic rounding (SR) process incurs significant hardware overhead

Solution/ Contribution

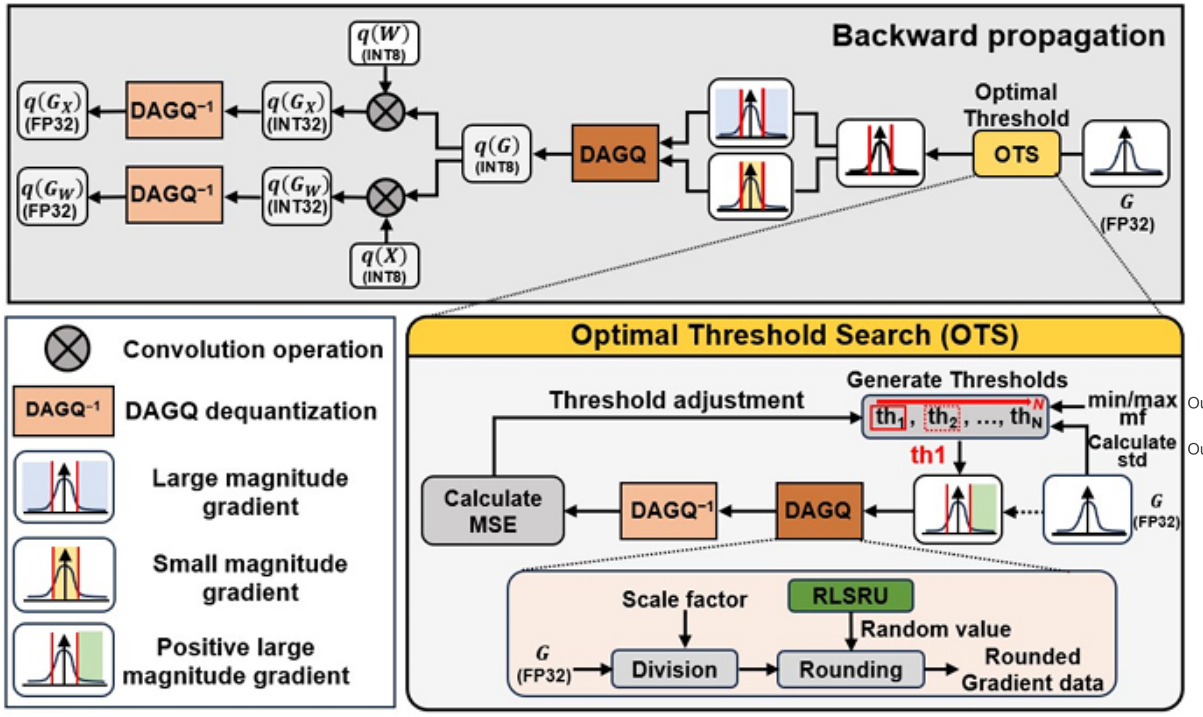
1 Dual-Scale Adaptive Gradient Quantization (DAGQ)

Effectively capture each region's characteristics and apply different scale factors based on a threshold (from OTS) between small and large magnitude gradients

2 Reusable LFSR-based Stochastic Rounding Unit (RLSRU)

Effectively generate random numbers for SR using 4-bit linear feedback shift registers

Dataflow of gradient distribution-aware INT8 quantization

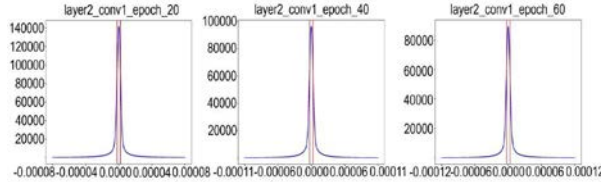


Ours : SR using PyTorch-based random number generation library

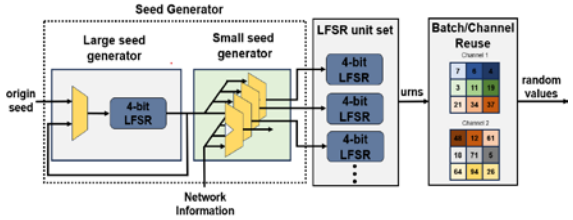
Ours* : SR implementation using RLSRU

Gradient distributions of ResNet18

Utilize kernel density estimation to analyze gradient distributions



Overview of the RLSRU framework



Top-1 accuracies on the ImageNet dataset

Model	Method	FP32(%)	INT8(%)	ACC drop(%)
ResNet18	DAI8 [3]	70.22	70.21	-0.01
	SRU-Q [5]	69.52	69.04	-0.48
	ESRU [4]	71.10	70.91	-0.19
	Ours	69.76	69.94	+0.18
	Ours*	69.76	69.50	-0.26
ResNet34	DAI8 [3]	73.46	73.40	-0.06
	Ours	73.30	73.79	+0.49
	Ours*	73.30	73.11	-0.19
ResNet50	DAI8 [3]	76.50	76.59	+0.09
	ESRU [4]	77.59	77.56	-0.03
	Ours	76.14	76.65	+0.51
	Ours*	76.14	75.85	-0.29
Mobile NetV2	DAI8 [3]	72.44	71.92	-0.52
	Ours	72.84	73.08	+0.24
	Ours*	72.84	72.35	-0.49

Resource Comparison with different LFSRs

Method	LFSR Bit-width	LUTs	FFs
SRU-Q [5]	12bit	4,423	5,596
ESRU [4]	3bit	16,384	32,768
Ours	4bit	542	785

"LowGradQ: Adaptive Gradient Quantization for Low-Bit CNN Training via Kernel Density Estimation-Guided Thresholding and Hardware-Efficient Stochastic Rounding Unit," DATE 2025 (BK21 Top-tier Conf.)

Lightweight Test-Time Adaptation for On-Device AI

IDSL Intelligent Digital Systems Design Lab.
SEOULTECH 서울과학기술대학교

Goal

Most memory-efficient and fastest TTA with high accuracy

Motivation

- Existing entropy minimization strategy requires both forward and backward passes through the entire model and introduces training delays and additional sub-processes due to low entropy data filtering

Solution/ Contribution

High-performance TTA achieved solely through stem layer redesign

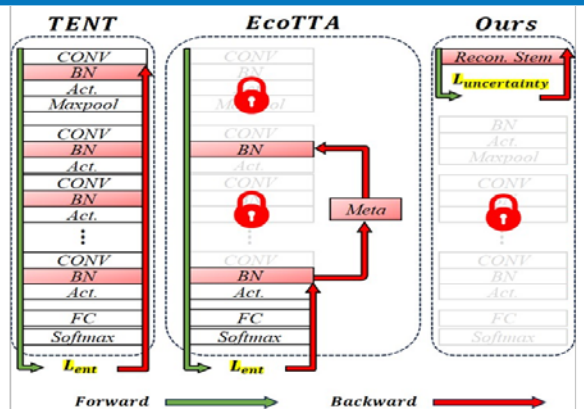
1 Gaussian Channel Attention Layer (GCAL)

Remodel the squeeze and excitation block to predict uncertainty and minimize the uncertainty instead of entropy

2 Domain Embedding Layer (DWT+IDWT)

DWT is applied to split the input into LFC/HFC, enhancing GCAL; then inverse DWT restores the original shape

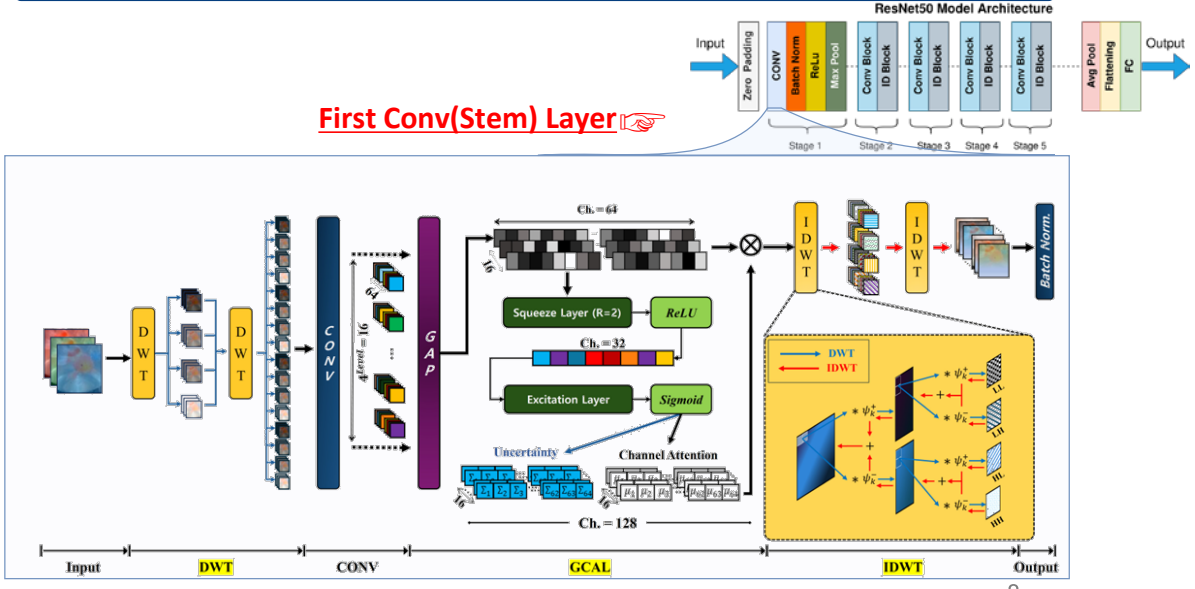
Diagram comparing the forward/backward flow and update process



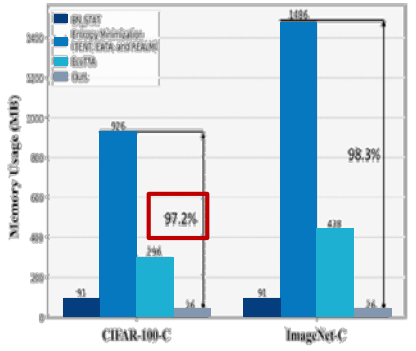
Prediction error comparison w/ prior TTA methods

Method	Noise			Blur				Weather				Digital				Avg.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	
	CIFAR-10-C Acc. +5.4%↑ Compared to SOTA															
Source	86.9	82.6	81.8	11.4	50.2	18.9	9.1	16.4	26.4	18.4	7.1	24.5	22.8	64.0	28.3	36.6
Ours w/o GCAL	62.4	55.4	48.5	12.1	51.7	15.5	8.5	16.6	26.4	21.7	6.4	31.0	21.3	61.7	22.5	30.8
TENT[59]	39.4	38.8	47.9	19.9	45.0	23.2	20.6	28.1	32.1	24.5	16.1	26.7	32.4	30.6	35.5	30.7
MONO[65]	43.5	39.9	43.3	26.4	44.4	25.1	25.0	20.9	28.3	22.8	11.9	28.3	21.1	42.8	21.7	29.7
SFT[33]	31.9	26.7	**28.9**	17.7	44.2	18.4	20.2	20.8	23.4	20.7	13.9	25.4	24.5	21.9	25.1	24.2
EATA[45]	33.9	32.8	41.4	19.4	42.4	20.5	20.1	22.4	27.1	22.7	13.8	24.0	24.0	29.3	26.5	26.7
SAR[46]	46.4	40.9	50.1	20.2	47.0	21.7	20.8	22.9	29.5	23.9	13.8	25.5	24.3	39.5	27.4	30.3
REALM[53]	**27.8**	**25.4**	35.5	15.5	37.7	17.4	16.9	20.4	22.3	19.0	12.9	18.0	23.1	22.0	24.2	22.5
Ours w/o DEL	33.1	29.1	31.3	10.0	36.9	**11.4**	7.3	**12.8**	**13.1**	**11.6**	**5.7**	**6.9**	**18.4**	15.7	24.5	17.8
Ours	31.0	26.8	30.5	**9.2**	**34.9**	**11.4**	**7.1**	13.1	13.8	12.2	**5.7**	7.5	**18.4**	**12.6**	22.6	**17.1**

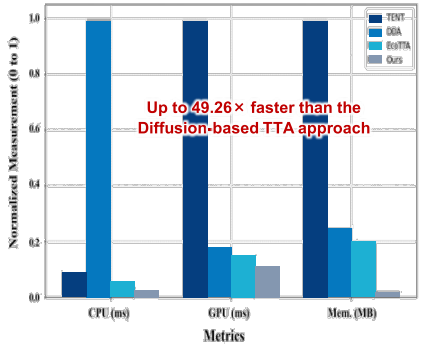
Overview of L-TTA including the reconstructed stem layer



Comparison of memory usage in a single iteration



Training time comparison of various TTA approaches



"L-TTA: Lightweight Test-Time Adaptation Using a Versatile Stem Layer," NeurIPS 2024 (BK21 Top-tier Conf.)

HW-Friendly PTQ for CNN-Transformer Hybrid Networks

Goal

Hardware-efficient lossless integer quantization of HybridViT

Motivation 1

- Massive outliers in the post-pointwise convolution activation
  → **Accuracy degradation**

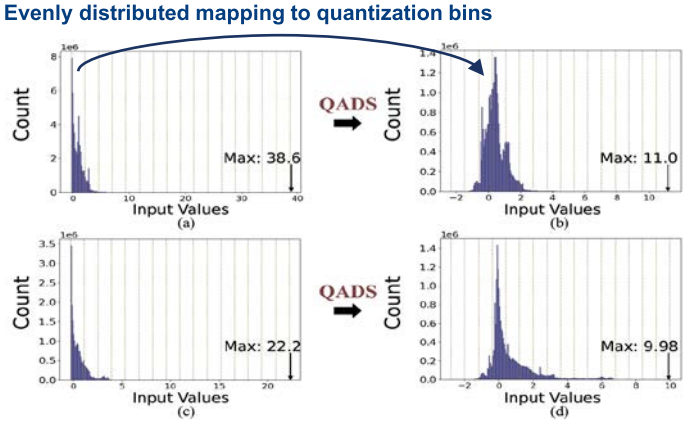
Motivation 2

- Quantization impossible due to non-linear operations in Exponential and Softmax
  → **Hardware-inefficient floating-point operations**

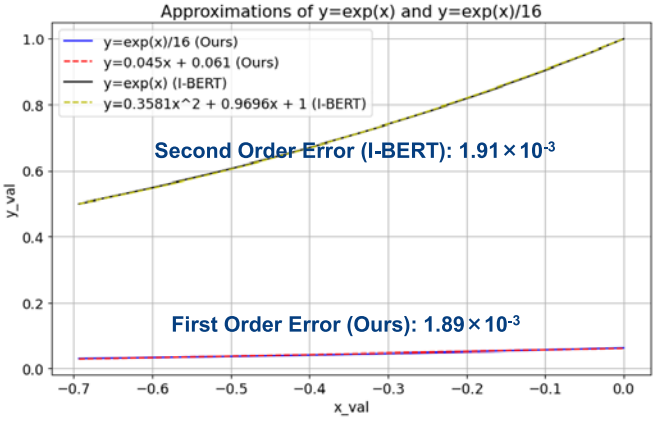
Solution/ Contribution

1. **Quantization-Aware Distribution Scaling (QADS)**
   Outlier reduction through MSE-based optimization method
2. **Linear Softmax (LinMax)**
   Linear approximation of Softmax exponential to enable integer operations
   → **All operations including Softmax processed in int8 (fully-integer)**

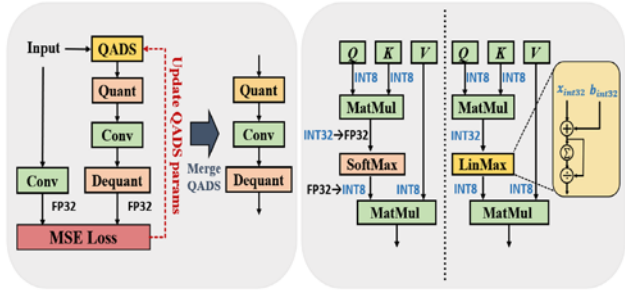
Activation histograms in MobileViT MBConv (a,c): before QADS. (b,d): after QADS



Comparison between the approximated second-order and proposed first-order polynomial



Overview of HyQ. (a): QADS optimization. (b): Attention using linear Softmax



Comparison of accuracy with other methods

Model	Method	Prec. (W/A)	Size	Top-1 Acc.	Acc. Drop
Mobile ViT-xxs	Baseline	32/32	1.27MB	68.94%	-
	FQ-ViT	8/8	0.32MB	66.46%	2.48%
	Percentile	8/8	0.32MB	67.54%	1.40%
	OMSE	8/8	0.32MB	66.69%	2.25%
	Q-HyViT	8/8	0.32MB	67.20%	1.74%
	Ours	8/8	0.32MB	68.15%	0.79%
Mobile ViT-xs	Baseline	32/32	2.32MB	74.63%	-
	FQ-ViT	8/8	0.58MB	68.28%	6.35%
	Percentile	8/8	0.58MB	62.96%	11.67%
	OMSE	8/8	0.58MB	68.41%	6.22%
	Q-HyViT	8/8	0.58MB	73.89%	0.75%
	Ours	8/8	0.58MB	73.99%	0.64%
Mobile ViT-s	Baseline	32/32	5.58MB	78.32%	-
	FQ-ViT	8/8	1.40MB	77.67%	0.65%
	Percentile	8/8	1.40MB	77.85%	0.47%
	OMSE	8/8	1.40MB	77.61%	0.71%
	Q-HyViT	8/8	1.40MB	77.72%	0.59%
	Ours	8/8	1.40MB	77.93%	0.39%

Comparison of hardware resource utilization

Method	Unroll	LUT	FF	DSP	Power(mW)
Ours / I-BERT	16	2096 / 3776	1068 / 2096	0 / 48	251 / 371
	32	4237 / 7597	2087 / 3306	0 / 96	324 / 569
	64	8391 / 17676	4131 / 5573	0 / 174	450 / 988

"HyQ: Hardware-Friendly Post-Training Quantization for CNN-Transformer Hybrid Networks," IJCAI 2024 (BK21 Top-tier Conf.)

Integrating Low-Rank Approximation and Quantization for ViTs

Goal

A vision transformer compression framework for efficient inference

→ **LRA-QViT framework effectively integrates low-rank approximation (LRA)** and **quantization** to enhance inference efficiency of ViTs on edge and mobile devices

Motivation 1

- Lower reconstruction error from LRA improves the potential for achieving higher accuracy during fine-tuning

Motivation 2

- Applying the proposed RB-LRA amplifies activation outlier effects, making existing distribution scaling methods suboptimal for accuracy

Solution/ Contribution

1 Error Compensation Matrix Design for LRA

Design a reparameterizable error matrix in the form of LRA using a residual branch(RB-LRA) as follows

$$y \approx V'(U'^T x) + \hat{E}x = \underbrace{(V' + \tilde{V})(U'^T + \tilde{U}^T)}_{\text{Reparameterizable!}}x \quad \text{where } \hat{E}x = (V'\tilde{U}^T + \tilde{V}U'^T + \tilde{V}\tilde{U}^T)x$$

Reparameterization

$$y \approx (V' + \tilde{V})(U'^T + \tilde{U}^T)x = A'B'^T x, \quad A' \in \mathbb{R}^{r \times n}, B' \in \mathbb{R}^{m \times r}, r \ll \frac{mn}{m+n}$$

2 Weight Reconstruction (WR)

RB-LRA Initialization method (Initialization using removed elements from LRA → Minimize reconstruction error)

$$\begin{cases} U_{\tilde{V}} = U_{[:,r:]} \\ S_{\tilde{V}} = S_{[r:,r:]} \\ V_{\tilde{V}} = V_{[r:,:]} \end{cases}$$

LRA Deleted Matrices

$$y' = U^T x = \boxed{Concat(U'^T x, U_{\tilde{V}}^T x)}$$

Concat Reconstruction : $\tilde{U} \rightarrow Zero$ init.

$$y = VS^T y' = \boxed{V' y'_{[r:,:]} + (S_{\tilde{V}} V_{\tilde{V}})^T y'_{[r:,:]}}$$

Residual Reconstruction : $\tilde{V} \rightarrow (S_{\tilde{V}} V_{\tilde{V}})^T$ init.

RB-LRA & WR: $\|w - (w_r + \hat{E})\|_F^2$ < LRA: $\|w - w_r\|_F^2 = \sum_{i=r+1}^{min(m,n)} s_i^2$

Eckart-Young-Mirsky

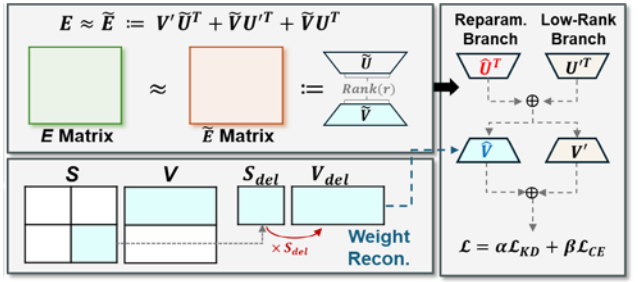
$$\|w - w_r\|_F^2 = \left\| \sum_{i=r+1}^{min(m,n)} s_i u_i v_i^T \right\|_F^2 = \sum_{i=r+1}^{min(m,n)} s_i^2$$

3 Weight-Aware Distribution Scaling (WADS)

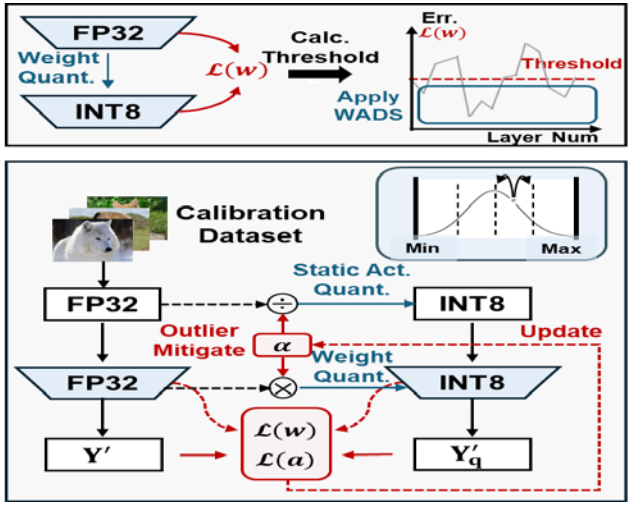
Activation outliers are migrated to the weights via the weight quantization regularization loss term

$$a' = \underset{\alpha}{\operatorname{argmin}} \boxed{\left\| Q\left(\frac{x}{\alpha}\right) Q(\alpha w) - xw \right\|^2} + \boxed{\|Q(w) - w\|^2}$$

Overview of RB-LRA & WR



Overview of WADS



Inference Speedup on Edge & Mobile Devices

Model	Method	Prec.	Size(MB)	Android(ms)	Xavier(ms)
DeiT-B	Baseline	FP32	346.4	275.6	150.7
	RB-LRA	FP32	177.6	153.2	73.6
	RB-LRA+WADS	INT8	44.4 (87.2%)	86.7 (3.18x)	59.4 (2.54x)
Swin-T	Baseline	FP32	113.2	98.5	61.1
	RB-LRA	FP32	84.4	83.6	38.6
	RB-LRA+WADS	INT8	21.1 (81.4%)	67.3 (1.46x)	27.4 (2.23x)
Swin-B	Baseline	FP32	352.4	287.4	140.5
	RB-LRA	FP32	240.4	226.3	102.2
	RB-LRA+WADS	INT8	60.1 (83.0%)	155.3 (1.85x)	96.2 (1.46x)

"LRA-QViT: Integrating Low-Rank Approximation and Quantization for Robust and Efficient Vision Transformers," ICML 2025 (BK21 Top-tier Conf.)

Robust and Efficient Gradient Quantization for ViTs

Goal

A Vision transformer gradient quantization framework for stable training

The **GradQ-ViT framework** effectively mitigates outliers that cause quantization errors based on convergence theory, enabling stable training

Motivation

- Outlier cause large quantization error → **Training becomes unstable**

Solution/ Contribution

1 Cross-Huber Blend Loss (CH-Loss)

Design a new loss function that combines the cross-entropy (CL) and Huber (HL) functions to mitigate outliers in the loss

2 Interquartile range (IQR)-Driven Quantization Strategy (IQS)

Adaptively assigning quantization points by analyzing the characteristics of the gradient distribution based on IQR

3 Gradient Scaling for Efficient Dequantization (GS)

The quantized gradient is adjusted to have the same magnitude and direction as the original gradient

4 Adaptive Learning Rate Allocation (ALA)

Adaptively assigning the learning rate during training based on the layer's quantization error and cosine similarity

Accuracy results of DeiT&SwinT family model

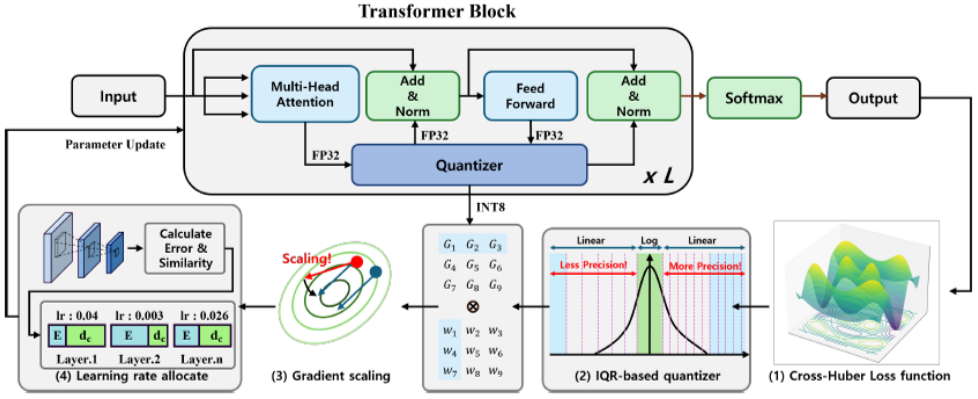
Model	Method	Baseline (%)	Precision (F/B)	Accuracy (%)	Drop (%)
DeiT-T	SR	72.2	8/8	57.75	14.45
	Quantformer	72.2	4/32	69.9	2.3
	Q-VIT	72.86	4/32	72.79	0.07
	Ours	**72.2**	**8/8**	**72.21**	**-0.01**
DeiT-S	SR	79.8	8/8	69.52	10.28
	Quantformer	79.9	4/32	78.2	1.7
	Q-VIT	79.92	4/32	80.11	-0.19
	Ours	**79.8**	**8/8**	**80.26**	**-0.46**
DeiT-B	SR	81.8	8/8	74.38	7.42
	Quantformer	81.8	4/32	79.7	2.1
	Ours	**81.8**	**8/8**	**82.32**	**-0.52**

Model	Method	Baseline (%)	Precision (F/B)	Accuracy (%)	Drop (%)
Swin-T	SR	81.2	8/8	65.89	15.31
	Quantformer	81.2	4/32	78.3	2.9
	Q-VIT	80.9	4/32	80.59	0.31
	Ours	**81.2**	**8/8**	**81.15**	**0.05**
Swin-S	SR	83.2	8/8	69.83	13.37
	Quantformer	83.2	4/32	81	2.2
	Ours	**83.2**	**8/8**	**83.37**	**-0.17**
Swin-B	SR	84.5	8/8	72.73	11.77
	Ours	**84.5**	**8/8**	**84.71**	**-0.21**

Accuracy results of MobileViT

Model	Method	Baseline(%)	Accuracy(%)	Drop(%)
MobileViT-XXS	SR / Ours	69.76	56.25 / 69.58	13.51 / **0.18**
MobileViT-XS	SR / Ours	75.21	61.74 / 75.07	13.47 / **0.14**
Mobile ViT-S	SR / Ours	79.05	66.02 / 78.96	13.03 / **0.09**

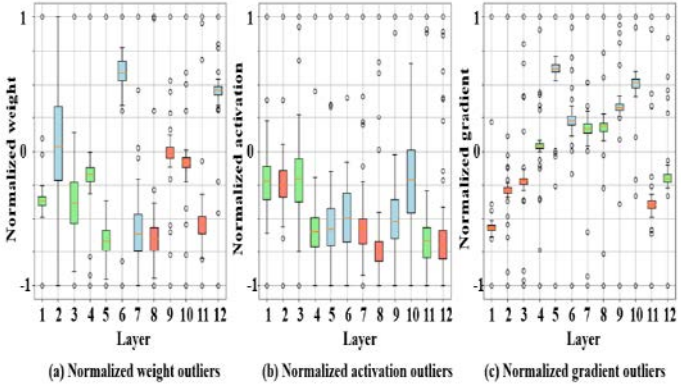
Overview of GradQ-ViT framework



Running time of MobileViT

Precision (F/B)	Forward (ms)	Backward (ms)	Iteration (ms)
32/32	**84.37**	**180.42**	**281.45**
8/32	41.19	**180.42**	240.73
4/32	27.83	**180.42**	206.72
8/8	**41.19**	**79.38**	**136.58**
	2.04×	**2.27×**	**2.06×**

Visualization of the W/A/G outlier in the DeIT-Tiny model



"GradQ-ViT: Robust and Efficient Gradient Quantization for Vision Transformers," AAAI 2025 (BK21 Top-tier Conf.)

Post-Training Quantization for Large-Scale ASR Models

Goal

A mixed-precision post-training quantization framework for efficient and accurate deployment of large-scale ASR models

Motivation 1

- Quantization-aware training achieves high accuracy but requires prohibitive retraining cost, limiting practicality for ASR models

Motivation 2

- Mixed-precision quantization improves the model size–accuracy trade-off, but existing methods suffer from large search space

Solution/ Contribution

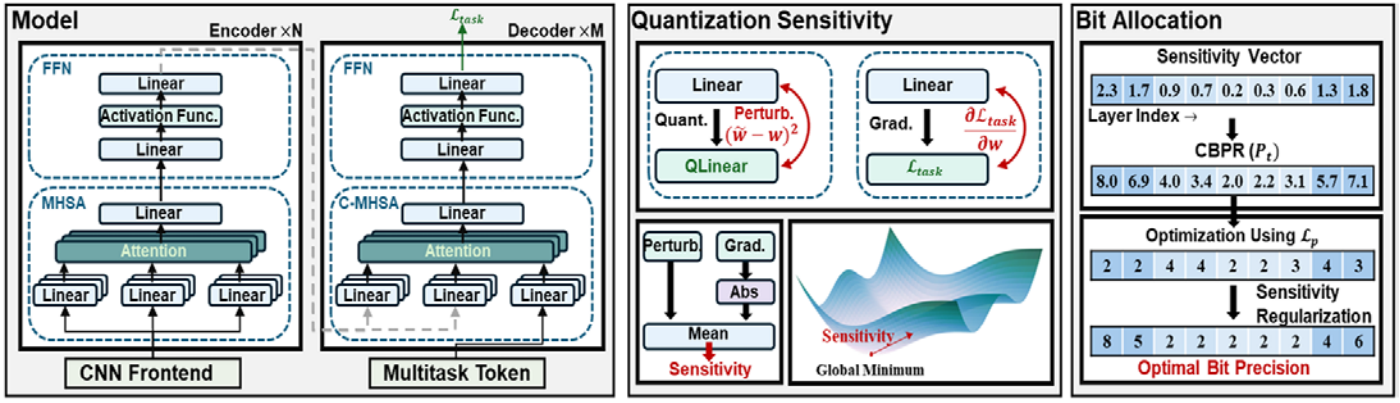
1 Gradient-based sensitivity metric

Quantifies layer-wise importance by combining quantization perturbation with gradient information, guiding optimal bit allocation

2 Continuous bit-precision representation with sensitivity regularization

Transforms discrete bit-precision into differentiable variables for fast optimization, while preventing over-aggressive reduction in sensitive layers

Overview of Proposed GenPTQ Architecture



ASR Accuracy of Conformer-Large

Dataset	Method	Prec.	WER(↓)	CER(↓)
Voxpopuli	Baseline	FP32	6.9	3.8
		INT8	6.9	3.9
	MinMax	INT4	7.3	4.1
		INT2	15.3	6.8
	Ours	**INT2.5**	**7.8**	**4.3**
MLS-German	Baseline	FP32	4.9	1.6
		INT8	4.9	1.6
	MinMax	INT4	5.0	1.7
		INT2	36.5	12.6
	Ours	**INT2.5**	**5.7**	**1.8**
MLS-Spanish	Baseline	FP32	4.4	1.8
		INT8	4.5	1.8
	MinMax	INT4	4.8	1.9
		INT2	54.2	39.3
	Ours	**INT2.5**	**5.4**	**2.1**

Optimization Cost–Accuracy Trade-off

Method	Prec.	WER(↓)	CER(↓)	Optimization Time
Baseline	FP32	4.1	1.6	-
AdaRound	INT2	5.8	2.4	8h 10m
BRECQ		5.5	2.5	2h 41m
Ours	**INT2.5**	**4.3**	**1.7**	**15.16s**

Speech Recognition Accuracy on Whisper

Method	Prec.	Size (MB)	Libri speech		Common Voice		Giga Speech		SPGI Speech	
			WER	CER	WER	CER	WER	CER	WER	CER
Baseline	FP32	2955.9	4.1	1.6	11.4	5.5	12.6	7.5	3.7	1.8
MinMax	INT8	864.5	4.2	1.6	11.5	5.5	12.8	7.6	3.7	1.8
	INT4	489.2	4.2	1.7	11.7	5.6	12.7	7.6	3.8	1.9
	INT2	301.5	15.5	9.8	28.7	16.2	43.4	37.0	19.2	13.2
Ours	**INT2.5**	**323.2**	**4.3**	**1.7**	**13.8**	**6.6**	**12.8**	**7.6**	**4.4**	**2.1**

Multilingual Speech Recognition Accuracy on Whisper

Method	Prec.	Size (MB)	MLS-German		MLS-Portuguese		MLS-Spanish		MLS-Polish	
			WER	CER	WER	CER	WER	CER	WER	CER
Baseline	FP32	2955.9	6.8	2.4	8.7	3.1	5.1	1.6	6.5	1.4
MinMax	INT8	864.5	6.7	2.3	8.9	3.2	5.1	1.6	6.6	1.4
	INT4	489.2	6.9	2.4	9.2	3.6	5.2	1.7	6.8	1.6
	INT2	301.5	51.7	39.8	64.5	38.4	30.4	16.4	69.7	54.5
Ours	**INT2.5**	**323.2**	**7.9**	**3.1**	**10.3**	**3.6**	**6.3**	**2.0**	**7.9**	**1.9**

"GenPTQ: Green Post-Training Quantization for Large-Scale ASR Models with Mixed-Precision Bit Allocation," EMNLP 2025 (Findings) (BK21 Top-tier Conf.)

Efficient KV Cache Compression in LLMs

Goal

To alleviate the KV cache memory and accelerate LLM inference via a similarity-aware, multi-level cache management framework

Motivation 1

- During LLM inference, the KV cache grows linearly with sequence length, necessitating the reduction of memory usage

Motivation 2

- Previous studies mainly relied on attention heuristics for cache compression, overlooking inter-layer/head dynamics

Solution/ Contribution

1 Multi-level redundancy scoring

Quantifies representational redundancy across layers/heads using distance-based similarity,

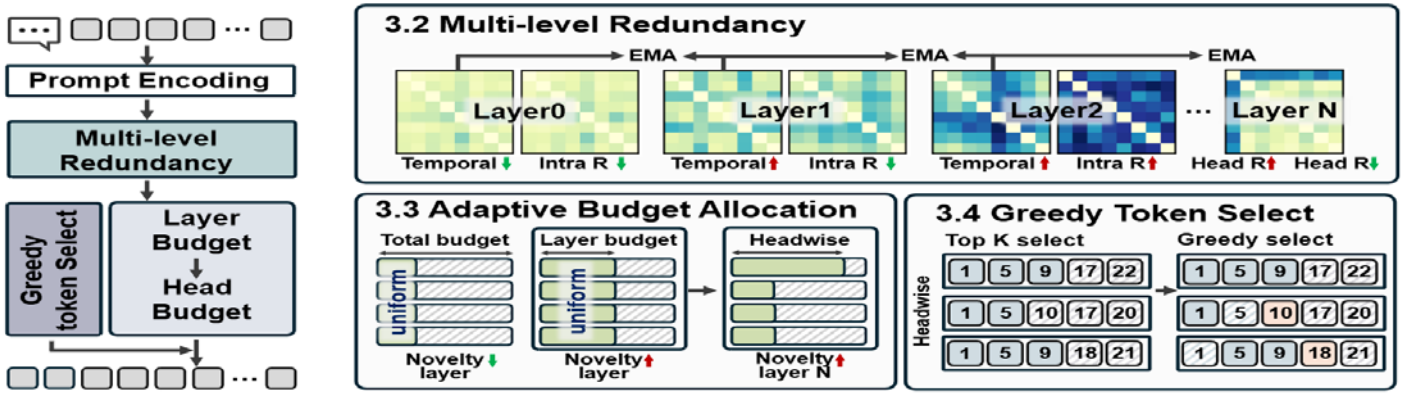
2 Adaptive budget allocation

Dynamically allocates cache budgets across layers and heads based on quantified redundancy

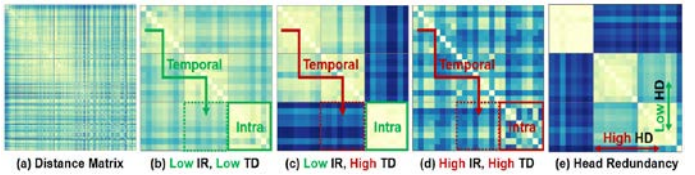
3 Redundancy-aware greedy token selection

Selects informative and diverse tokens through a redundancy-aware greedy algorithm

An overall architecture of SCORE framework



Head-wise cosine distance matrice



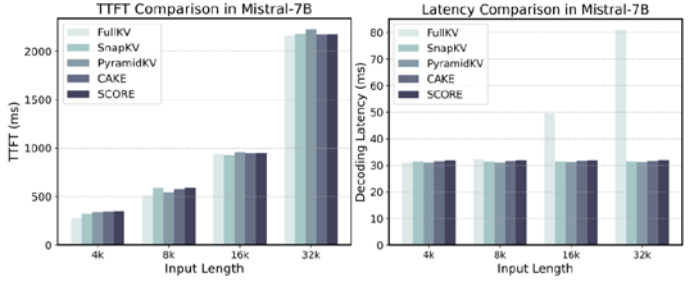
Comparison on Llama3-8B-Instruct over LongBench V2

Method	Easy	Hard	Short	Medium	Long	Avg.
Full KV	31.25	25.08	34.44	24.19	22.22	27.44
Llama3-8B-Instruct, Cache size=128						
CAKE	29.27	21.13	**32.98**	21.22	20.18	24.96
HeadKV	29.53	22.74	32.25	22.97	20.10	25.52
SCORE	**29.94**	**23.11**	32.78	**23.73**	**20.33**	**25.98**
Llama3-8B-Instruct, Cache size=1024						
CAKE	29.50	21.31	33.25	22.61	21.06	25.55
HeadKV	30.22	23.97	33.24	23.23	20.14	26.16
SCORE	**30.25**	**24.08**	**33.89**	**24.19**	**21.45**	**26.77**

Reasoning results on Mistral-7B-Instruct with 128 Budget

Method	0k	1k	2k	4k	8k	16k	32k	Avg.
Full KV	61.30	55.30	53.40	42.10	40.30	34.00	31.80	45.46
SnapKV	55.40	50.20	46.40	37.20	35.00	32.80	29.20	40.89
PyramidKV	57.20	50.80	47.60	36.20	36.20	31.40	28.20	41.09
HeadKV	**58.60**	53.80	**52.20**	38.20	37.60	31.80	30.40	43.23
CAKE	58.40	54.00	51.30	**38.40**	37.20	31.80	30.20	43.04
SCORE	58.40	**54.20**	51.80	38.30	**37.80**	**32.10**	**30.60**	**43.31**

Comparison of TTFT (left) and decoding latency (right)



"SCORE: Similarity-Aware Contextual Overlap-Redundancy Eviction for Efficient KV Cache Compression in LLMs," ICLR 2025, Submitted

Hybrid Attention for Long Context LLMs

Goal

Design a training-free attention that combines linear attention and KV-cache eviction for efficient long-sequence LLM inference

Motivation

- Linear attention reduces complexity but requires retraining; KV-cache eviction avoids retraining but causes accuracy loss
- Existing eviction methods ignore value impact, leading to poor cache allocation and degraded accuracy

Solution/ Contribution

1 Taylor Approximation Based Train Free Linear Attention

Adapt a dynamically scaled 1st-order Taylor approximation to Softmax to achieve training-free linear attention

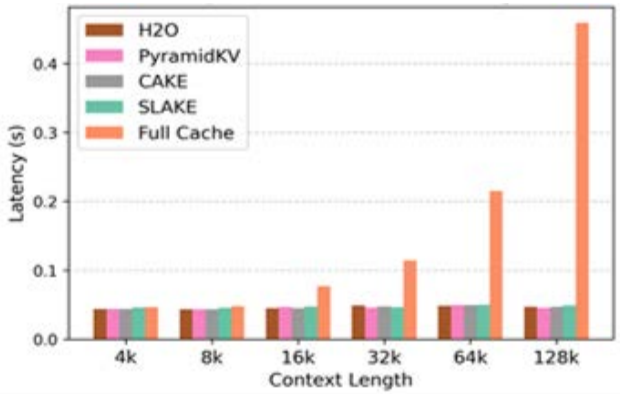
2 PTAA (Partially Taylor Approximated Attention

Combine first-order Taylor approximated linear attention with eviction based Softmax attention enabling accurate train-free approximation while preserving pretrained weights

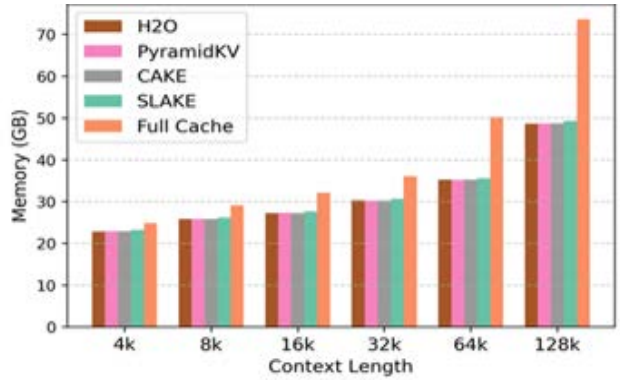
3 VABS (Value-Aware Budget Scoring)

Allocate cache budget dynamically by considering both approximation error and Value impact, improving attention fidelity compared to entropy-based scoring

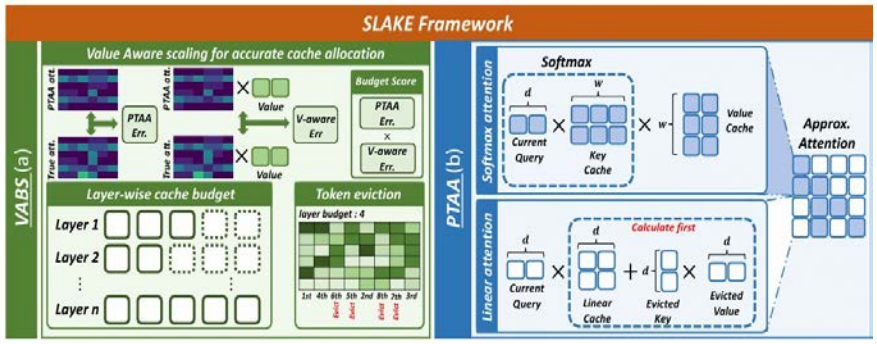
Latency for Different Context Length



Memory Usage for Different Context Lengths



Overall architecture of proposed Hybrid Attention



Performance evaluation with existing KV Cache Eviction Methods

Model	Method	Cache Size	Single-Document QA			Multi-Document QA			Summarization			Few-shot Learning			Synthetic		Code		Avg.
			NtvQ	Qspr	MF	HtptQ	2WkMQ	Msq	GvRpr	QMS	MltNs	TREC	TrvQA	SMSm	PCnt	PR-en	Lcc	RB-P	
llama2-7B chat	Full KV	-	18.74	21.41	37.57	27.78	32.03	7.53	26.85	20.97	26.41	64.00	83.59	41.37	2.85	7.00	60.48	54.39	33.31
	H2O	128.00	14.23	14.78	26.51	20.05	28.73	5.00	16.20	20.30	20.64	38.00	72.47	37.10	2.30	3.50	49.69	47.37	26.05
	PyramidKV	128.00	13.45	16.61	32.52	22.76	28.59	7.56	18.87	19.96	20.05	43.50	79.90	36.74	2.29	8.00	51.19	47.56	28.10
	cake	128.00	14.27	16.50	31.87	24.27	27.08	7.60	19.11	20.64	20.63	47.00	80.53	37.83	3.25	8.00	54.11	50.25	28.93
	SLAKE	128.00	**13.94**	**15.60**	**31.24**	**25.84**	**30.77**	**7.73**	**19.64**	**20.35**	**20.61**	**54.50**	**80.60**	**37.85**	**3.24**	**8.50**	**55.29**	**50.52**	**29.76**
	H2O	256.00	15.27	15.31	27.24	21.02	27.34	6.31	19.75	20.45	22.23	45.51	79.64	37.93	2.93	4.00	52.45	51.23	28.04
	PyramidKV	256.00	15.13	15.86	33.93	25.58	29.51	9.23	20.35	21.22	22.01	58.00	82.39	38.47	2.15	7.50	55.81	49.53	30.42
	CAKE	256.00	15.38	15.62	35.55	26.92	30.04	9.18	20.39	20.92	22.34	58.00	81.97	38.91	2.72	6.50	58.03	52.48	30.93
	SLAKE	256.00	**16.36**	**15.69**	**35.79**	**27.77**	**30.77**	**8.92**	**20.40**	**20.71**	**22.61**	**61.50**	**82.08**	**39.04**	**2.46**	**7.50**	**58.87**	**52.36**	**31.43**

"SLAKE : Softmax-Approximated Training-Free Linear Attention With KV-Cache Eviction For Long-Sequence LLMs," ICLR 2026, Submitted

Importance-Aware Token Dropping with Local Coverage Guarantee in VLMs

Goal

Design of an efficient VLM inference acceleration framework by jointly preserving salience and diversity while guaranteeing local coverage

Motivation

- Existing token-dropping methods rely on attention scores or redundant pivots, leading to overhead, position bias, and local coverage loss, thereby limiting efficient and accurate VLM inference under tight token budgets.

Solution/ Contribution

1 Temporal-shift salience estimation

Estimate token importance by measuring hidden-state variations across ViT layers, providing an attention-free salience signal that avoids positional bias and extra computation

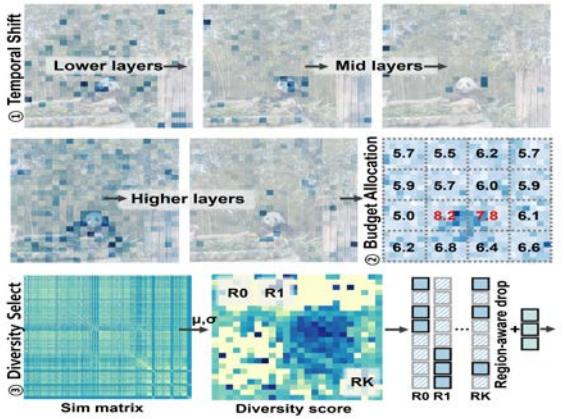
2 Adaptive region-level budget allocation

The image is partitioned into spatial regions, and token budgets are dynamically assigned based on regional salience to provide more consistent behavior, ensuring balanced and stable local coverage even under tight token budgets.

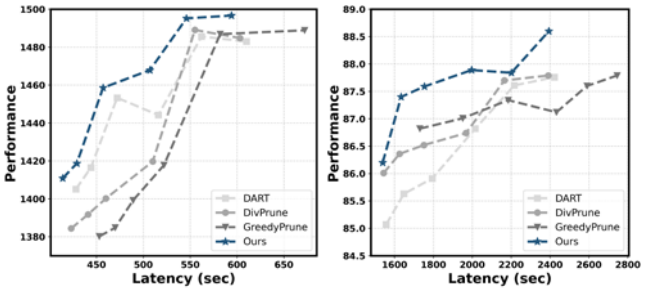
3 Diversity score-based selection

An efficient similarity statistic is used to suppress redundancy and select representative tokens, maintaining diverse visual information while minimizing computation.

Proposed SPLIT-VLM processing pipeline



Latency–performance comparison



Comparative results of LLaVA-Next-7B on 10 image understanding tasks

Method	Image Understanding: 10 Vision-Language Model Benchmark Tasks							Avg.
	GQA	MMB	MMB-CN	MME	POPE	SQA	VQA^{txt}	
LLaVA-Next-7B *Upper Bound, All Tokens* (100%)								
Vanilla	62.2	66.3	57. 0	1483.5	87.7	69.3	69.3	100.0%
LLaVA-Next-7B *Tokens Reduction* (↓ 66.7%)								
DivPrune	61.5	66.0	55.4	1470.6	87.6	**69.9**	58.7	98.9%
DART	61.2	66.0	56.1	1473.6	**87.7**	69.2	**62.6**	99.3%
GreedyPrune	60.9	**66.1**	54.2	1457.3	86.8	68.6	57.3	97.9%
SPLIT (Ours)	**61.7**	**66.1**	**56.2**	**1479.3**	87.5	**69.9**	**62.6**	**99.6%**
LLaVA-Next-7B *Tokens Reduction* (↓ 88.9%)								
DivPrune	59.8	64.9	52.6	1384.4	86.0	**68.8**	50.0	93.5%
DART	**60.5**	65.1	53.4	**1416.5**	85.6	67.8	**57.5**	95.6%
GreedyPrune	59.5	64.8	52.3	1383.8	85.4	67.8	50.4	93.3%
SPLIT (Ours)	60.3	**65.2**	**53.6**	1410.9	**86.2**	68.6	55.4	**96.0%**

LLaVA-Video-7B evaluation on video datasets

Method	LongVideoBench			Video-MME		Avg.
	Val	Perception	Relation	W/O Sub	Long	
LLaVA-Video-7B *Upper Bound, All Tokens* (100%)						
Vanilla	59.0	65.1	53.5	63.6	53.1	100.0%
LLaVA-Video-7B *Retain 64 × 64 Tokens* (↓ 62.1%)						
DivPrune	**58.7**	**64.2**	53.4	61.0	50.8	96.7%
DART	58.6	**64.2**	**53.5**	**61.5**	51.0	96.6%
GreedyPrune	58.4	**64.2**	52.7	61.3	50.8	95.9%
SPLIT (Ours)	**58.7**	**64.2**	53.4	61.2	**51.1**	**96.9%**
LLaVA-Video-7B *Retain 64 × 16 Tokens* (↓ 90.5%)						
DivPrune	51.5	56.8	**46.2**	55.5	47.4	86.9%
DART	51.4	56.9	46.0	54.7	**47.6**	87.3%
GreedyPrune	50.9	55.8	46.0	55.0	47.2	86.1%
SPLIT (Ours)	**51.8**	**57.1**	46.1	**56.3**	**47.6**	**87.7%**

" SPLIT-VLM: Salience-Guided Partitioning Towards Local Coverage for Importance-Aware Token Dropping in Vision-Language Models," ICLR 2026, Submitted

Hardware-efficient Mixture-of-Experts Compression

Goal

To develop a hardware-efficient MoE compression framework that preserves expert features and runs on a single GPU.

Motivation

- Although only a few experts are used during inference, MoE models must keep all expert weights in memory, causing severe memory overhead and latency on single-GPU systems.

Solution/ Contribution

1 Clustering-based Base–Delta and Truncated SVD

Clusters experts by cosine similarity, creates a routing-weighted Base, and applies Truncated SVD only to residual Delta to reduce redundancy while preserving expressiveness.

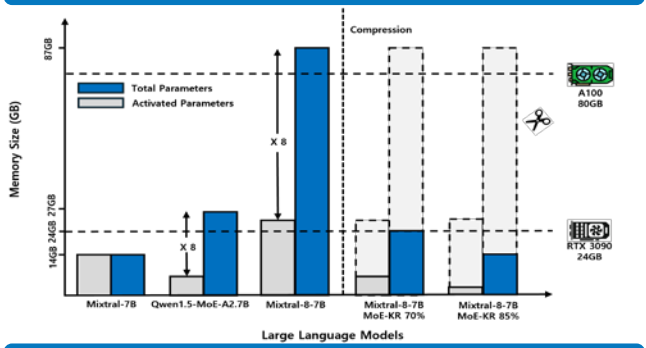
2 L2-Norm adapted Mixed-Precision Quantization

Assigns different bit-widths to μ (cluster base), B (shared basis), and u (expert coefficients) based on L2-norm importance, reducing memory usage while maintaining performance.

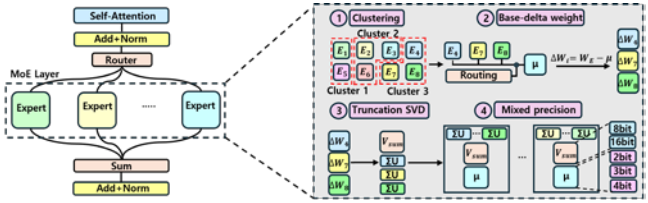
3 State-Of-The-Art Performance–Efficiency Trade-off

Records SOTA under 70–85% compression with lower PPL and faster latency.

Parameter size at mixtral-8-7B



Overview of MoE-KR framework



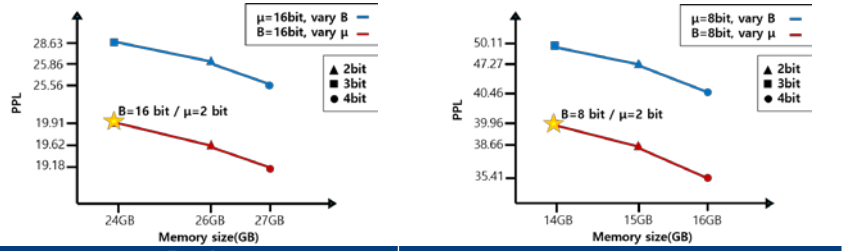
Comparison of MoE compression method at high compression ratio

Compression ratio	Method	WikiText-2 (PPL)	LAMBADA	PIQA	Memory	Latency	Token/s
			Mixtral-8-7B				
0%	Dense	9.05	62.50%	62.00%	87GB	9.45s	54.17
70%	ASVD	82916.53	0.00%	50.00%	24GB	6.90s	74.20
	NAEE	22.39	50.00%	49.50%	24GB	6.60s	77.25
	HC-MoE	21.66	50.50%	50.50%	24GB	5.31s	96.42
	MoE-KR	**19.91**	**51.50%**	**52.50%**	24GB	**4.49**	**114.03**
85%	ASVD	138275.89	0.00%	44.00%	14GB	4.39s	116.63
	NAEE	43.18	40.50%	45.50%	14GB	3.91s	130.95
	HC-MoE	42.48	41.00%	44.50%	14GB	3.35s	152.84
	MoE-KR	**40.26**	**42.50%**	**47.50%**	14GB	**2.73s**	**187.55**
Compression ratio	Method	WikiText-2 (PPL)	LAMBADA	PIQA	Memory	Latency	Token/s
			Qwen1.5-MoE-A2.7B				
0%	Dense	11.49	58.59%	60.00%	27GB	4.60s	111.30
70%	ASVD	98208.11	0.00%	46.50%	9GB	2.98s	171.81
	NAEE	27.23	45.50%	47.00%	9GB	2.65s	193.21
	HC-MoE	26.57	46.00%	46.50%	9GB	2.39s	214.23
	MoE-KR	**25.19**	**47.00%**	**48.00%**	9GB	**2.17s**	**236.41**
85%	ASVD	163996.75	0.00%	0.00%	4GB	2.12s	241.51
	NAEE	51.64	37.50%	42.50%	4GB	1.93s	265.80
	HC-MoE	51.01	38.00%	43.00%	4GB	1.77s	289.83
	MoE-KR	**48.66**	**39.50%**	**45.50%**	4GB	**1.56s**	**328.21**

Mean L2 norm values of μ, B, and u for Mixtral-8×7B

Mean L2 norm	μ	B	u	PPL	LAMBAD	PIQA
K = 2, r = 1	132.13	227.31	0.57	14.52	55.50%	50.50%
K = 2, r = 1	139.64	258.22	0.68	13.86	56.00%	52.00%
K = 2, r = 1	146.34	282.49	0.81	10.41	59.00%	58.00%

Effect of mixed precision with μ, B, and u



μ, B = 8			μ, B = 16		
Quantization	PPL	Memory	Quantization	PPL	Memory
Dense	9.05	87GB	Dense	9.05	87GB
u = 2	28.91	22GB	u = 2	16.52	43GB
u = 3	27.91	22GB	u = 3	16.19	43GB
u = 4	27.38	22GB	u = 4	15.66	43GB

" MoE-KR: Hardware-efficient Mixture-of-Experts Compression via K-means Cluster and R-Rank Decomposition," ICML 2026, Submitted