Model Quantization for Efficient Computer Vision

Haotong Qin
17.10.2023

北京航空航天大學 BEIHANG UNIVERSITY ETH zürich

个人信息

秦浩桐　　1997/07　　神经网络量化压缩

教育经历

- 2015-2019　北京航空航天大學 BEIHANG UNIVERSITY　计算机学院 本科
- 2022-2023　ETH zürich　Computer Vision Lab 访问学生
- 2019-至今　北京航空航天大學 BEIHANG UNIVERSITY　计算机学院/沈元学院（实验班）博士

（实验室：导师：李未院士、刘祥龙教授）

研究经历

- 2018-2019　Microsoft　MSRA 实习研究员（明日之星项目）
- 2020-2020　Tencent　WXG 实习研究员（犀牛鸟精英人才项目）
- 2021-2023　ByteDance　AI-Lab 实习研究员

直博 4年在TPAMI、NeurIPS等顶级会议期刊发表27篇文章，一作14篇，被引990余次

北京航空航天大学 BEIHANG UNIVERSITY ETH zürich

主要荣誉奖励

- 2023 **DAAD AInet Fellowship** （全球29人，中国机构唯一）
- 2023 **KAUST AI新星** （全球28人，中国机构首次/唯一）
- 2022 **字节跳动奖学金** （全国10人）
- 2023 **国家奖学金** （博，三次获奖）
- 2021 **国家奖学金** （博，二次获奖）
- 2020 **国家奖学金** （博）
- 2022 **北航十佳博士研究生**
- 2021 北京广受关注学术论文
- 2021 华为奖学金
- 2021 腾讯犀牛鸟精英人才
- 2019 ICPC全国邀请赛金牌
- 2018 ICPC全国邀请赛金牌
- 2022 北航五四奖章提名
- 2019-22 北航学业一等奖学金

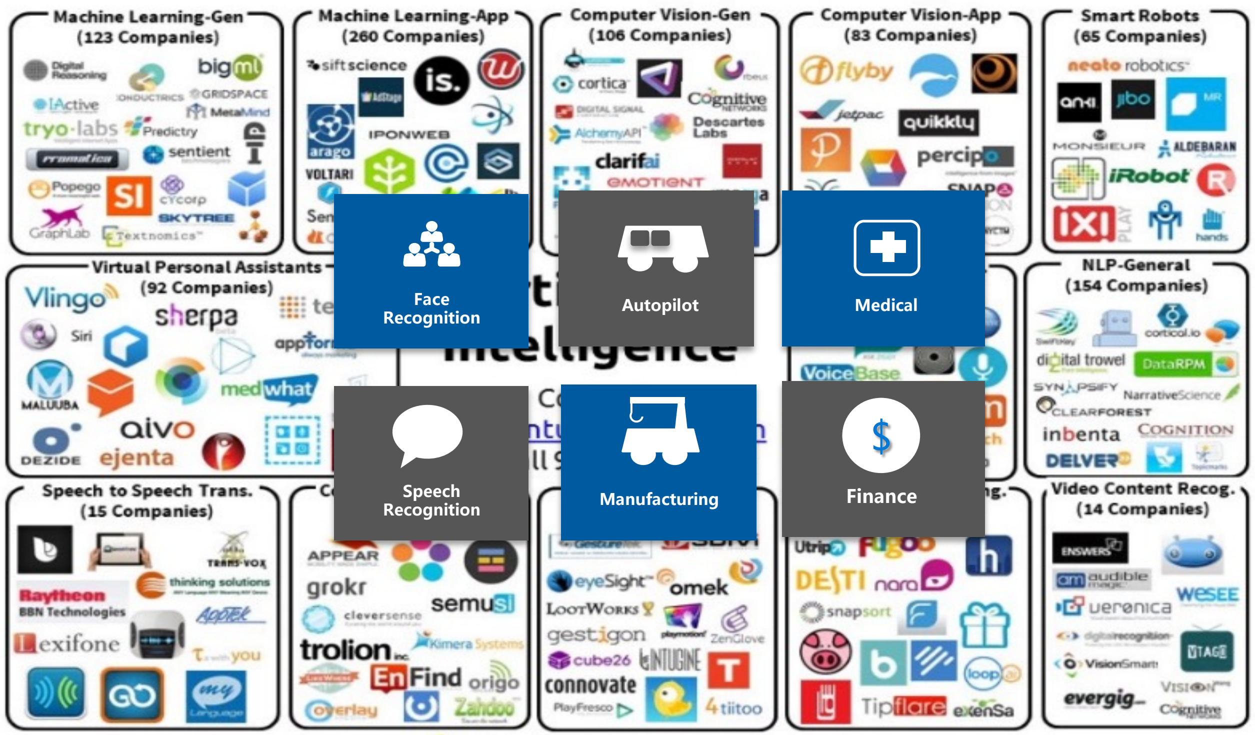
主要学术任职

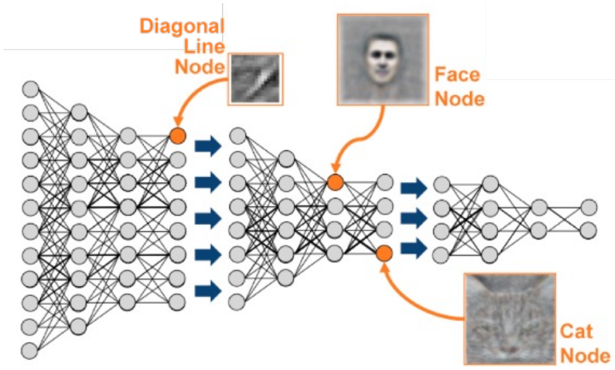
- **AAAI** （CCF-A） 2022/23 Workshop **组织者**
- **CVPR** （CCF-A） 2022/23 Workshop **竞赛主席**
- VALSE 2022 学生论坛 **组织者**
- PRCV 2021 专题论坛 **组织者**
- TPAMI/CVPR/ICCV等10余顶会顶刊 **审稿人**
- **CVPR 2023 Outstanding Reviewer (272/7000)**

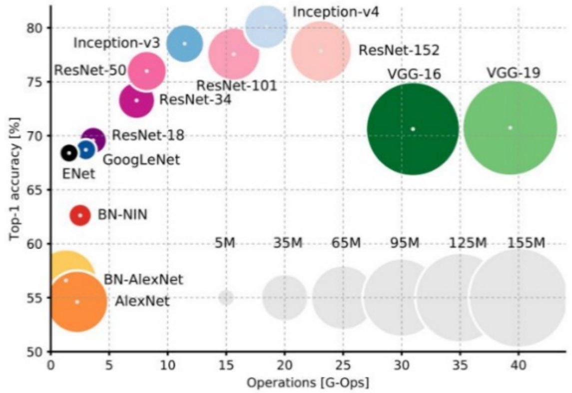
Background: deep learning and challenges

- **Vision**
  - Classification
  - Detection
  - Localization
  - Segmentation
- **Language**
  - Information retrieval
  - Relation extraction
  - Machine translation
- **Speech**
  - Language understanding
  - Speech recognition

…

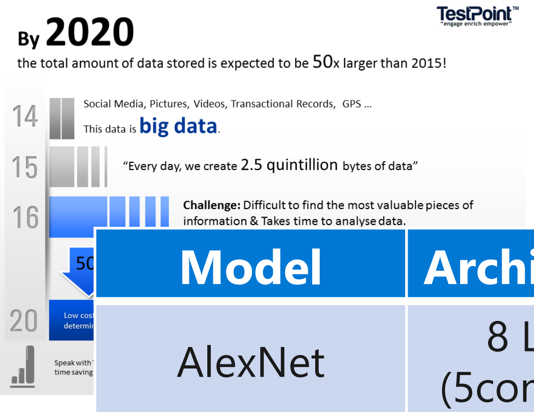


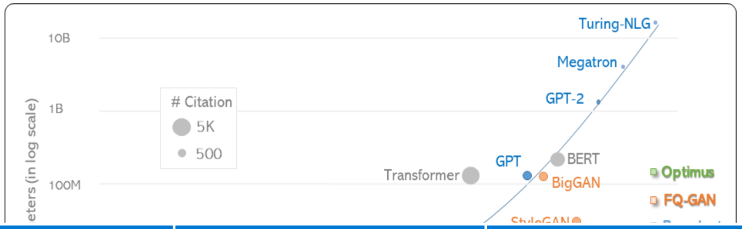




Background: deep learning and challenges

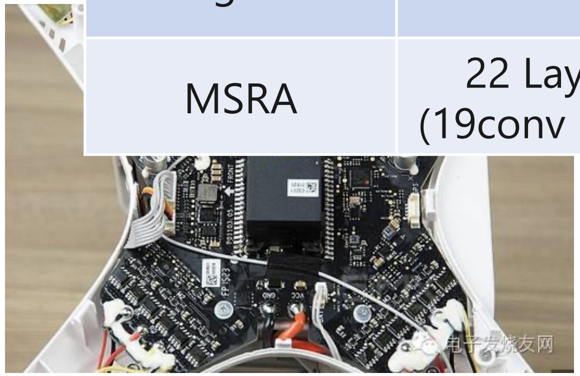
bigger data and larger model

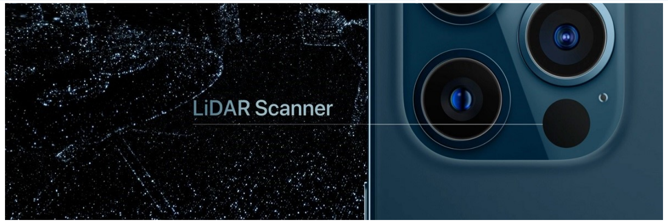


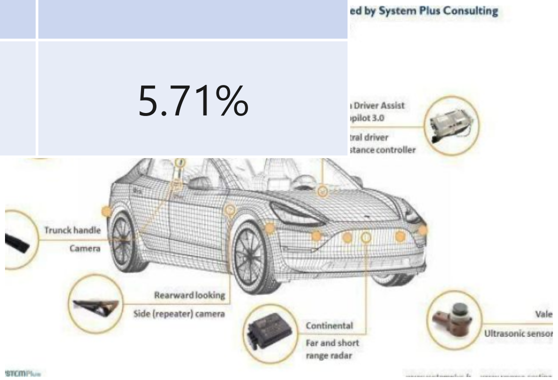


Model	Architecture	Parameters	Top-1 ERR	Top-5 ERR
AlexNet	8 Layers (5conv + 3fc)	~ 60 million	40.7%	15.3%
VGG	19 Layers (16conv + 3fc)	~ 144 million	24.4%	7.1%
GoogLeNet	22 Layers	~ 6.8 million	-	7.9%
MSRA	22 Layers (19conv + 3fc)	~ 200 million	21.29%	5.71%

diverse usage and limited resources







Background: model quantization

Quantization and Binarization

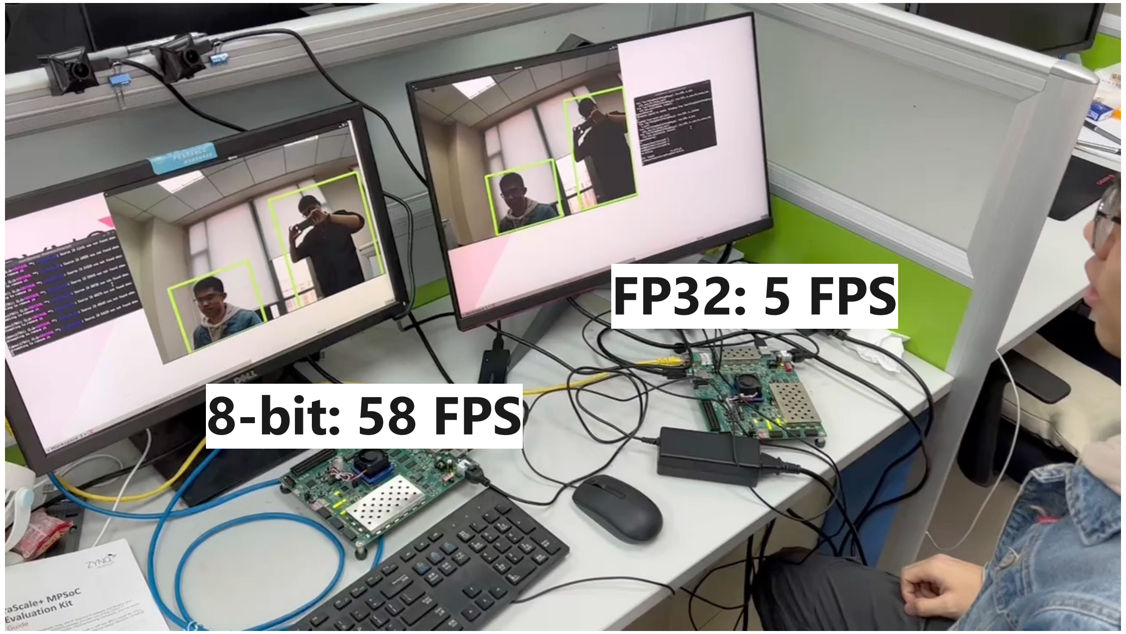
$$\underset{\text{32-bit}}{\mathbb{R}} \longrightarrow \underset{\text{8-bit}}{\mathbb{I}} \longrightarrow \underset{\text{1-bit}}{\mathbb{B}} \in \{-1, +1\}$$

Full-Precision Neural Networks

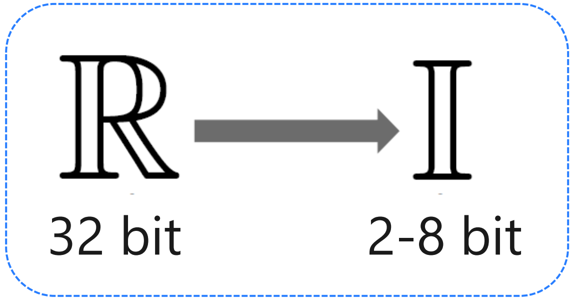
- Massive Parameters
- Complex Computation
- High Power Consumption

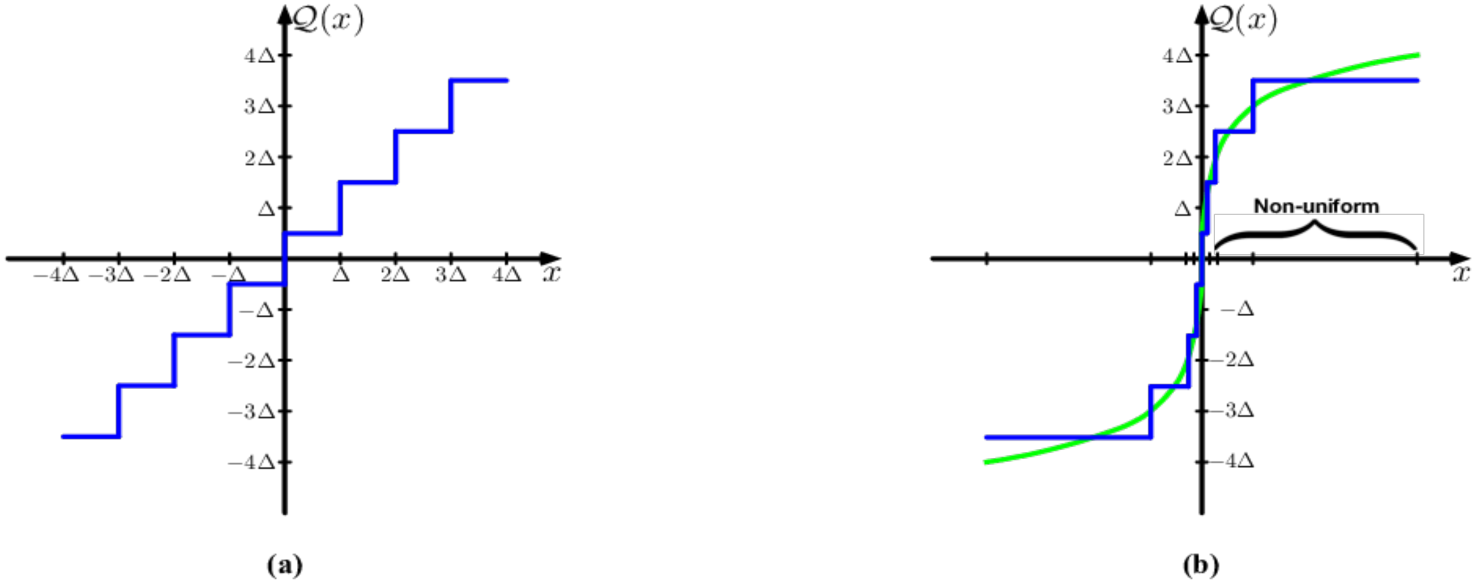
Low-Bit Quantized Neural Networks

- Quantized Parameters
- Efficient Instructions
- Low Power Consumption



Background: model quantization (2~8-bit)





Uniform Quantization

$$Q_U(x) = \mathrm{round}\left(\frac{x}{\Delta}\right)\Delta$$

$$\Delta = \frac{u-l}{2^b-1}$$

Non-Uniform Quantization

$$Q_U(x) = \mathrm{round}\left(\frac{\log_2 x}{\Delta}\right)\Delta$$

$$\Delta = \frac{u-l}{2^b-1}$$

Background: model binarization (1-bit)

1-Bit Parameters:

$$\mathbf{B_x} = \mathtt{sign}(\mathbf{x}) = \begin{cases} +1, & if\ \mathbf{x} \geq 0 \\ -1, & otherwise. \end{cases} \quad Q_x(\mathbf{x}) = \alpha \mathbf{B_x},$$

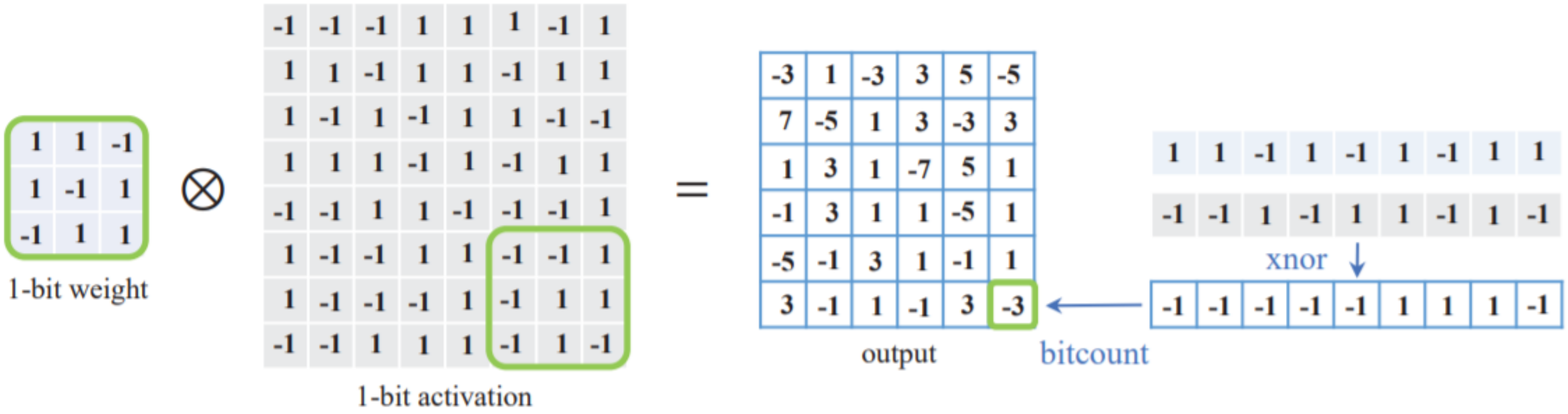
$$\mathbb{R} \longrightarrow \mathbb{B}$$

32 bit → 1 bit

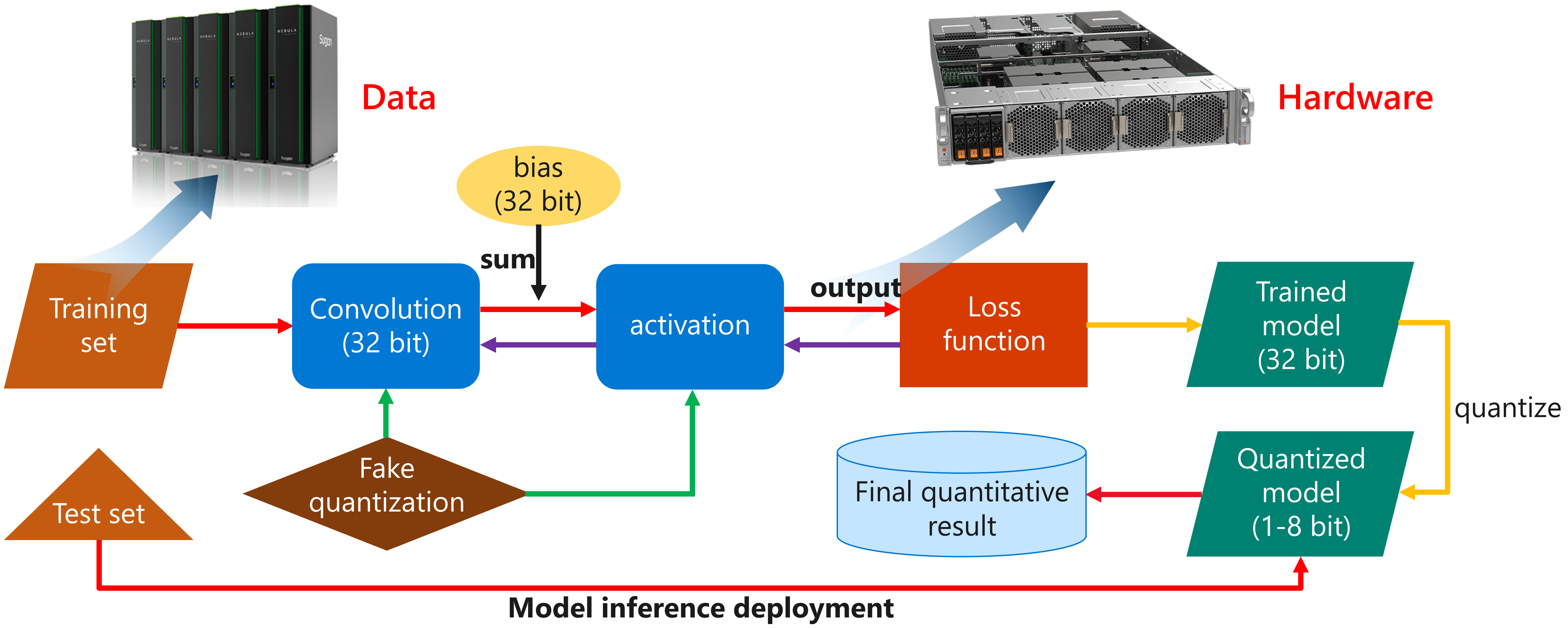
$\{-1, +1\}$

Bitwise Operations:

$$\mathbf{z} = \sigma(Q_w(\mathbf{w}) \otimes Q_a(\mathbf{a})) = \sigma(\alpha\beta(\mathbf{b_w} \odot \mathbf{b_a}))$$



Background: quantization pipeline



The most common pipeline of quantization is
training (fine-tuning) the 1-8 bit quantized models on the original dataset.

Binarization for Efficient Computer Vision

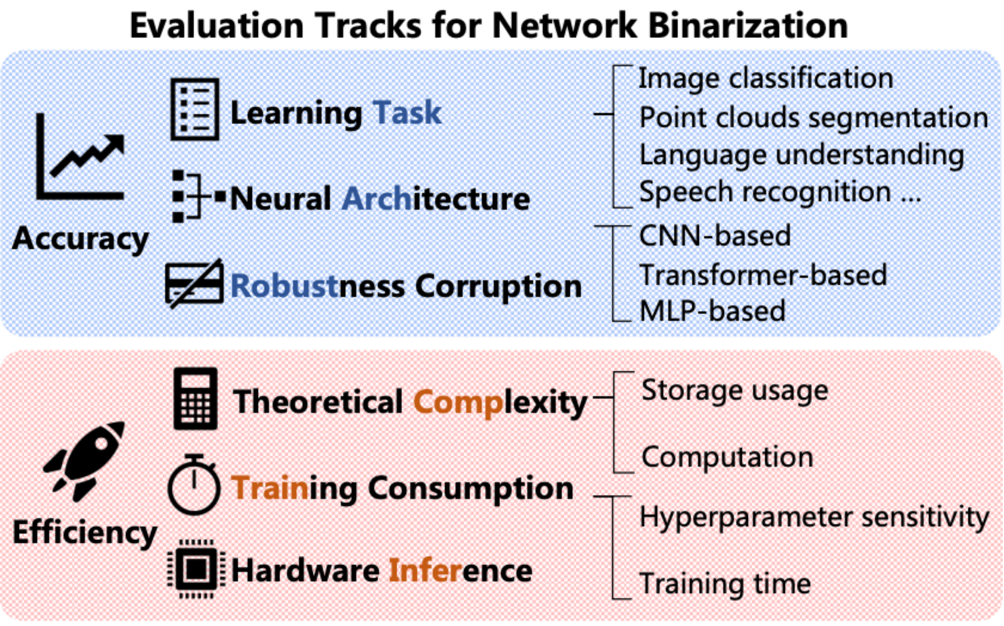
2. Binarization (1 bit)

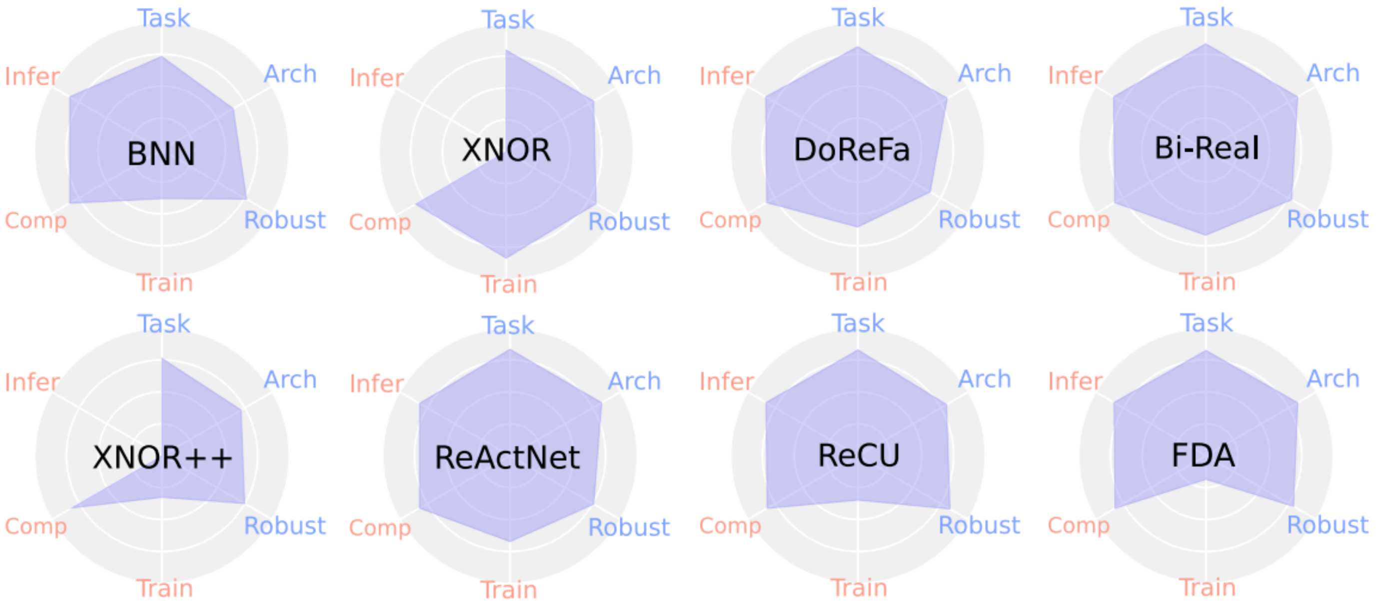
   2.1. BiBench: Benchmarking and Analyzing Network Binarization

   2.2. BiMatting: Efficient Video Matting via Binarization

   2.3. Flexible Residual Binarization for Image Super-Resolution

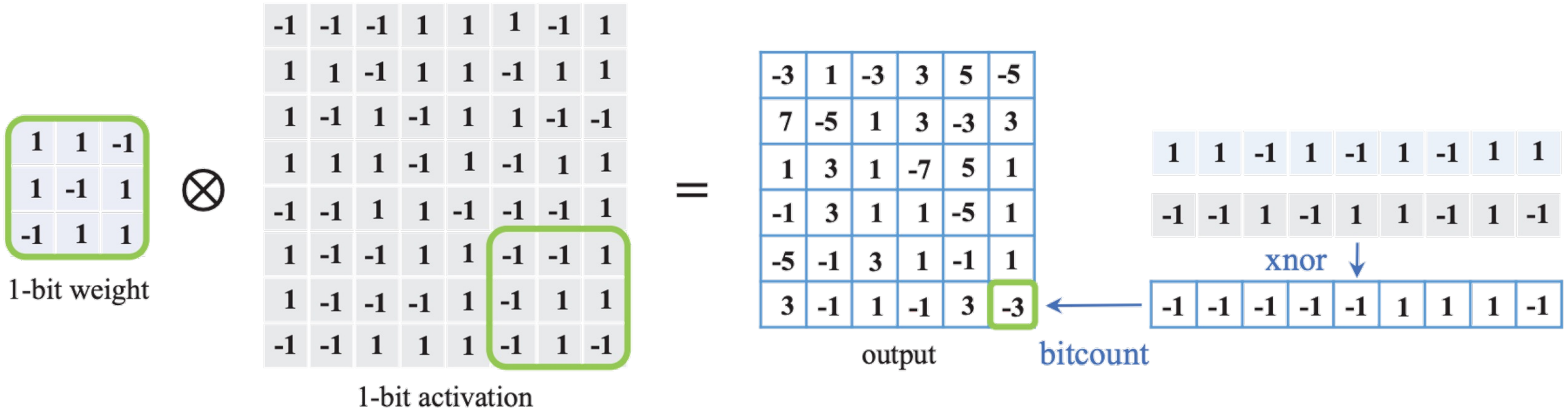
BiBench: Benchmarking and Analyzing Network Binarization





BiBench: model binarization

- *Network Binarization (1-bit)*



Equations: $Q_w(\boldsymbol{w}) = \alpha_w \mathbf{B}_w$ $\quad \mathbf{B}_{\boldsymbol{a}} = \mathrm{sign}(\boldsymbol{a}) = \begin{cases} -1, & \text{if } a \geq 0 \\ 1, & \text{otherwise} \end{cases}$ $\quad \boldsymbol{z} = Q_w(\boldsymbol{w})^\top \mathbf{B}_{\boldsymbol{a}} = \alpha_w (\mathbf{B}_{\boldsymbol{w}} \otimes \mathbf{B}_{\boldsymbol{a}})$

- Compressing neural networks by binarizing weights and activations
- Accelerating neural networks by applying bitwise instructions (e.g., XNOR and POPCNT)
- Theoretical acceleration and compression achieve 64x and 32x, respectively

BiBench: practical challenges of binarization

- *Trend-1: Accuracy comparison scope is limited*
  - Learning Task: most binarization algorithms to be only engineered for image inputs
  - Neural Architecture: monotonic tasks hinders a comprehensive evaluation for architectures
  - Corruption Robustness: data noise is hardly considered existing binarization algorithms
- *Trend-2: Efficiency analysis remains theoretical*
  - Theoretical Complexity: theoretical efficiency claims lack experimental evidence
  - Training Consumption: training efficiency of binarization algorithms is often ignored
  - Hardware Inference: the lack of hardware library support for deploying binarized models

BiBench: evaluation

- *Binarization Algorithms and Evaluation Tracks*
  - Binarization Algorithms:

Algorithm	Technique			Accurate Binarization			Efficient Binarization		
	s	τ	g	#Task	#Arch	Robust	Train	Comp	Infer
BNN (Courbariaux et al., 2016b)	×	×	√	3	3	*	√	√	√
XNOR (Rastegari et al., 2016)	√	×	×	2	3	*	√	√	√
DoReFa (Zhou et al., 2016)	√	×	×	2	2	*	×	√	×
Bi-Real (Liu et al., 2018b)	×	×	√	1	2	×	×	√	×
XNOR++ (Bulat et al., 2019)	√	×	×	1	2	×	×	×	×
ReActNet (Liu et al., 2020)	×	√	×	1	2	×	×	√	×
ReCU (Xu et al., 2021b)	×	√	√	2	4	×	×	×	×
FDA (Xu et al., 2021a)	×	×	√	1	6	×	×	×	×
Our Benchmark (***BiBench***)	√	√	√	**9**	**13**	√	√	√	√

  - Accuracy Tracks: evaluate accuracy of network binarization
  - Efficiency Tracks: evaluate efficiency of network binarization

BiBench: evaluation

- *Evaluation Metrics*
  - Learning Tracks:

$$\mathrm{OM}_{\text{task}}=\sqrt{\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}^2\left(\frac{\boldsymbol{A}^{bi}_{\text{task}_i}}{\boldsymbol{A}_{\text{task}_i}}\right)}$$

  - Neural Architecture:

$$\mathrm{OM}_{\text{arch}}=\sqrt{\frac{1}{3}\left(\mathbb{E}^2\left(\frac{\boldsymbol{A}^{bi}_{\text{CNN}}}{\boldsymbol{A}_{\text{CNN}}}\right)+\mathbb{E}^2\left(\frac{\boldsymbol{A}^{bi}_{\text{Transformer}}}{\boldsymbol{A}_{\text{Transformer}}}\right)+\mathbb{E}^2\left(\frac{\boldsymbol{A}^{bi}_{\text{MLP}}}{\boldsymbol{A}_{\text{MLP}}}\right)\right)}$$

  - Corruption Robustness:

$$\mathrm{OM}_{\text{robust}}=\sqrt{\frac{1}{C}\sum_{i=1}^{C}\mathbb{E}^2\left(\frac{\boldsymbol{G}_{\text{task}_i}}{\boldsymbol{G}^{bi}_{\text{task}_i}}\right)}$$

  - Theoretical Complexity:

$$\mathrm{OM}_{\text{comp}}=\sqrt{\frac{1}{2}\left(\mathbb{E}^2(\boldsymbol{r}_c)+\mathbb{E}^2(\boldsymbol{r}_s)\right)}$$

  - Training Consumption:

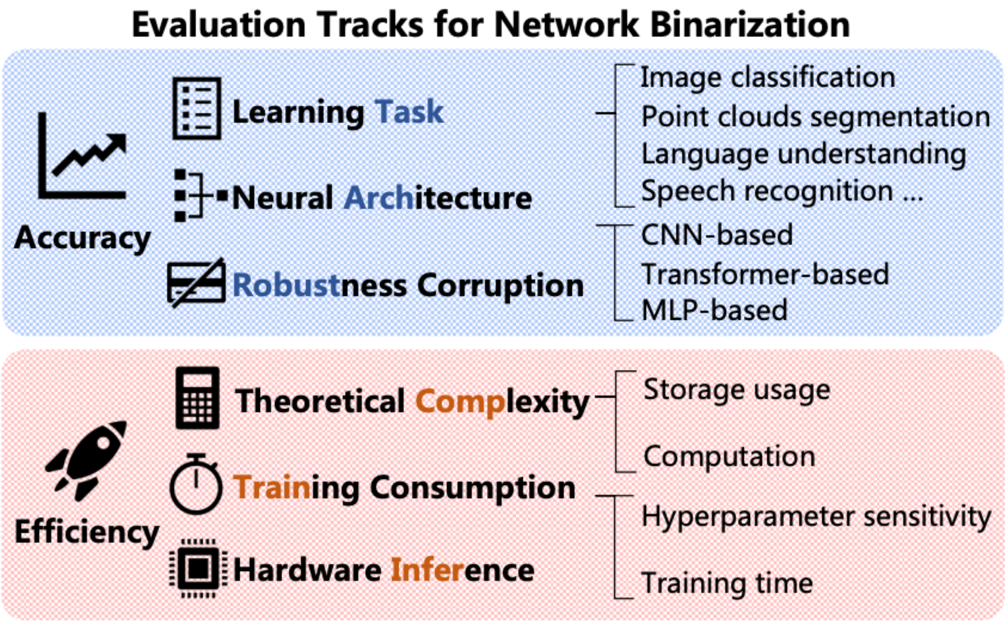
$$\mathrm{OM}_{\text{train}}=\sqrt{\frac{1}{2}\left(\mathbb{E}^2\left(\frac{\boldsymbol{T}_{\text{train}}}{\boldsymbol{T}^{bi}_{\text{train}}}\right)+\mathbb{E}^2\left(\frac{\text{std}(\boldsymbol{A}_{\text{hyper}})}{\text{std}(\boldsymbol{A}^{bi}_{\text{hyper}})}\right)\right)}$$

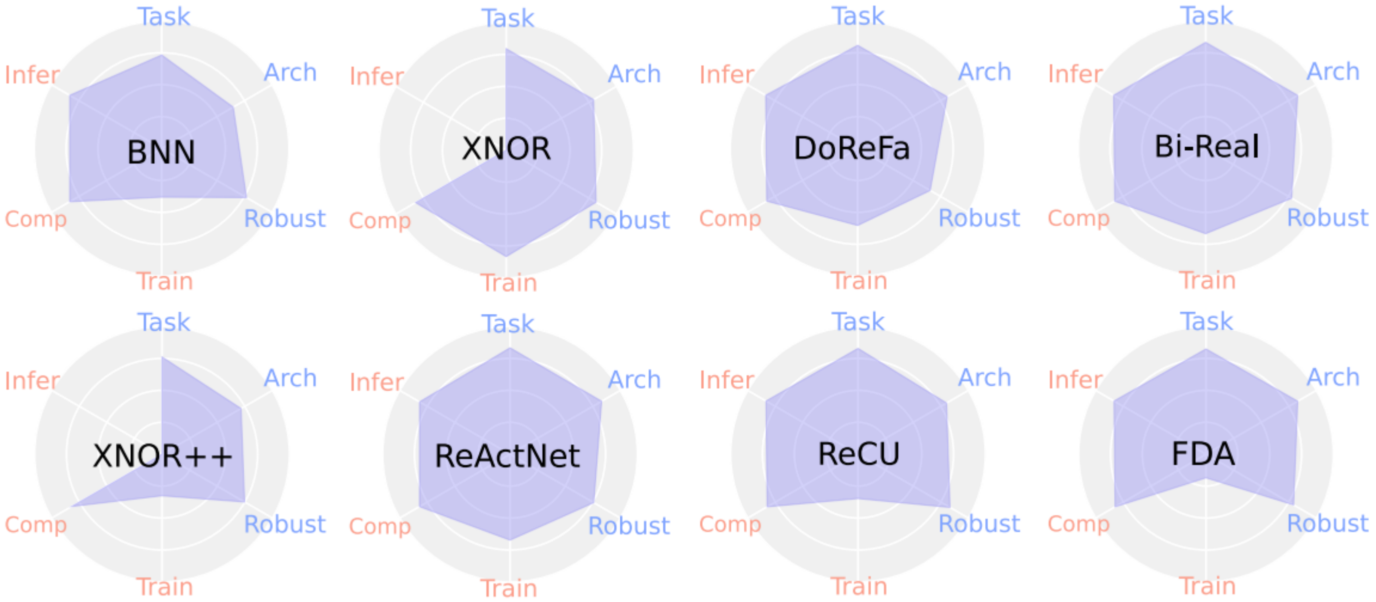
  - Hardware Inference:

$$\mathrm{OM}_{\text{infer}}=\sqrt{\frac{1}{2}\left(\mathbb{E}^2\left(\frac{\boldsymbol{T}_{\text{infer}}}{\boldsymbol{T}^{bi}_{\text{infer}}}\right)+\mathbb{E}^2\left(\frac{\boldsymbol{S}_{\text{infer}}}{\boldsymbol{S}^{bi}_{\text{infer}}}\right)\right)}$$

BiBench: evaluation

- *Performance*





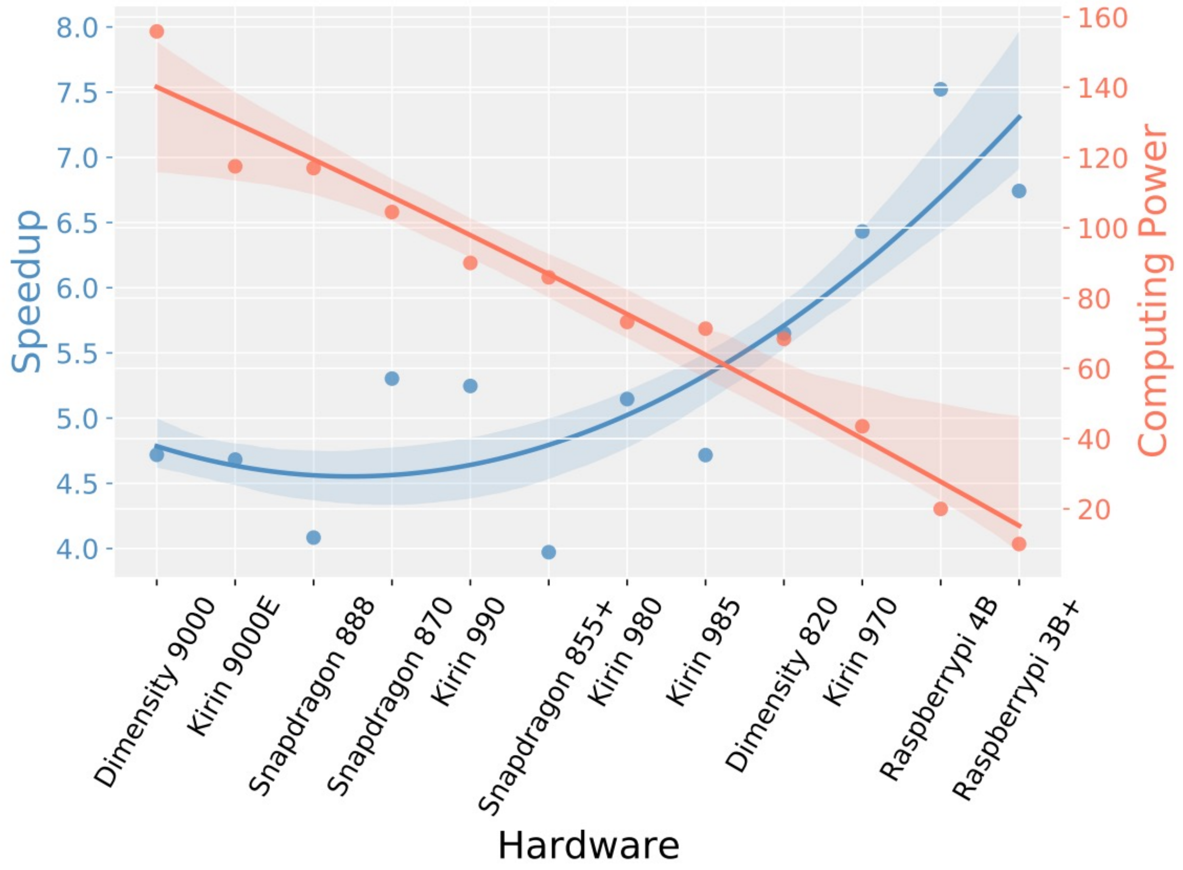
BiBench: analysis

- *Highlight Features:*
  - **1. Accuracy for Neural Architectures:** binarization exhibits a clear advantage on CNN- and MLP-based architectures compared to transformer-based ones
  - **2. Efficiency for Deployment Libraries:** limited inference libraries lead to almost fixed paradigms of binarization deployment

Infer. Lib.	Provider	s Granularity	s Form	Flod BN	Act. Re-scaling	Act. Mean-shifting
Larq	Larq	Channel-wise	FP32	√	×	√
daBNN	JD	Channel-wise	FP32	√	×	×
Algorithm	**Deployable**	**s Granularity**	**s Form**	**Flod BN**	**Act. Re-scaling**	**Act. Mean-shifting**
BNN	√	N/A	N/A	N/A	×	×
XNOR	×	Channel-wise	FP32	√	√	×
DoReFa	√	Channel-wise	FP32	√	×	×
Bi-Real	√	Channel-wise	FP32	√	×	×
XNOR++	×	Spatial-wise	FP32	×	×	×
ReActNet	√	Channel-wise	FP32	√	×	√
ReCU	√	Channel-wise	FP32	√	×	×
FDA	√	Channel-wise	FP32	√	×	×

BiBench: analysis

- *Highlight Features:*
  - **3. Born for Edge Hardware:** more promising for lower-power edge computing



BiBench: analysis

- *Suggested Paradigm of Binarization Algorithm*

(1) Soft gradient approximation (2) Channel-wise scaling factors (3) Pre-binarization parameter redistributing

Algorithm	Scaling Factor weight	Scaling Factor activation	Parameter Redistribution weight	Parameter Redistribution activation	Gradient Approximation weight	Gradient Approximation activation
BNN	w/o	w/o	w/o	w/o	STE	STE
XNOR	Statistics by Channel	Statistics by Channel	w/o	w/o	STE	STE
DoReFa	Statistics by Layer	w/o	w/o	w/o	STE	STE
Bi-Real	Statistics by Channel	w/o	w/o	w/o	STE	Differentiable Piecewise Polynomial Function
XNOR++	Learned by Custom-size $(o \times h_{\text{out}} \times w_{\text{out}})$	w/o	w/o	w/o	STE	STE
ReActNet	Statistics by Channel	w/o	w/o	w/o	STE	Differentiable Piecewise Polynomial Function
ReCU	Statistics by Channel	w/o	balancing (mean-shifting)	w/o	Rectified Clamp Unit	Rectified Clamp Unit
FDA	Statistics by Channel	w/o	w/o	mean-shifting	Decomposing Sign with Fourier Series	Decomposing Sign with Fourier Series

[1] “STE” indicates the Straight Through Estimator, and “w/o” means no special technique is used.

Binarization for Efficient Computer Vision

2. Binarization (1 bit)

   2.1. BiBench: Benchmarking and Analyzing Network Binarization

   2.2. BiMatting: Efficient Video Matting via Binarization

   2.3. Flexible Residual Binarization for Image Super-Resolution

Binarization for Efficient Computer Vision

BiMatting: Efficient Video Matting via Binarization

Compared to 1-bit video matting models using existing binarization methods, our BiMatting significantly surpasses them and achieves near full-precision performance. Note that the results of RVM-BNN indicate the model fully crashes.

BiMatting: motivation

- Running video matting application on edge

--**Through some lightweight video matting are proposed, their real-time inference still relies on expensive GPU device.**

--**Binarization is the most extreme bit-width compression technique, allowing model to utilize compact 1-bit parameter and efficient bitwise instructions.**

- Facing the challenge of accuracy drop

--**After binarization, the accuracy of model drops a lot, especially for the model with lightweight architecture (e.g., MobileNetV3 backbone).**

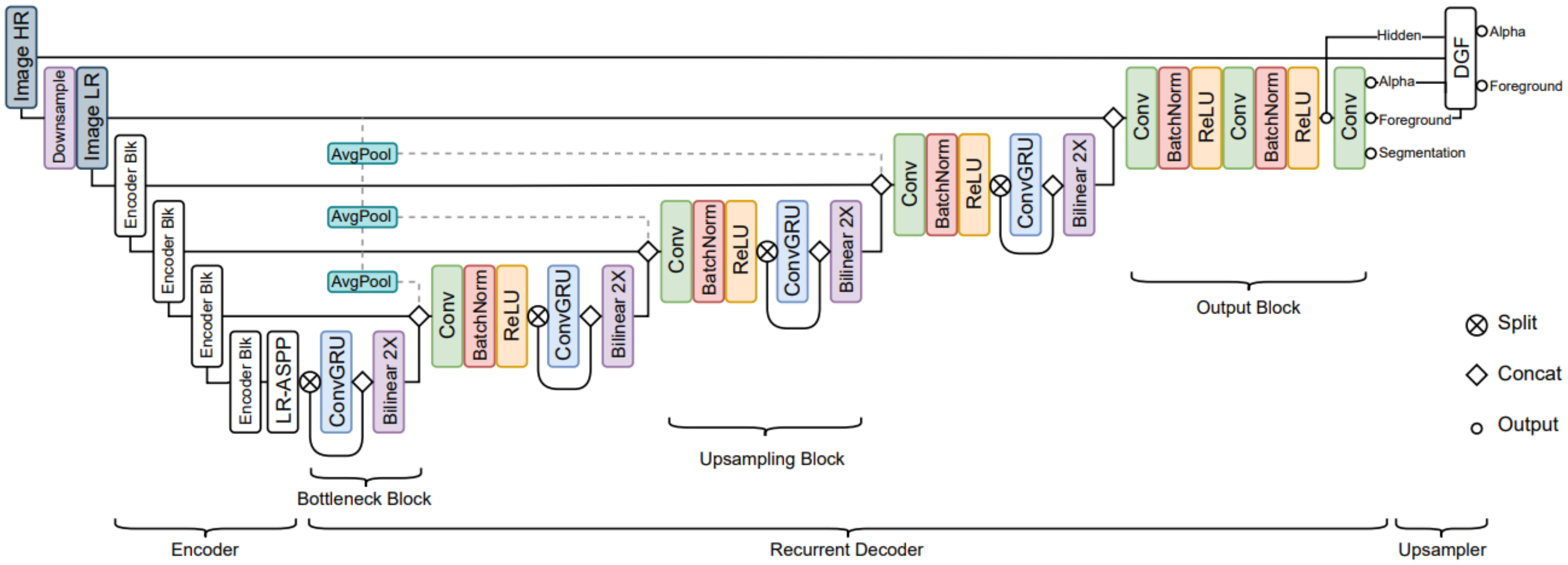
BiMatting: contribution

- We provide empirical studies of the accuracy and efficiency bottlenecks of matting binarization, and then propose BiMatting, a binarized model for accurate and efficient video matting.
- We propose **Shrinkable Binarized Block** (SBB), which follows a binarization-friendly computation-dense paradigm to construct a flexible block structure.
- We develop **Sparse-Assisted Binarization** (SAB) to effectively reduce the computational consumption of the binarized decoder.
- BiMatting achieves **12.4× FLOPs and 21.6× storage savings** compared to the full-precision counterpart, leading a promising way for the video matting on edge scenarios.

BiMatting: bottleneck

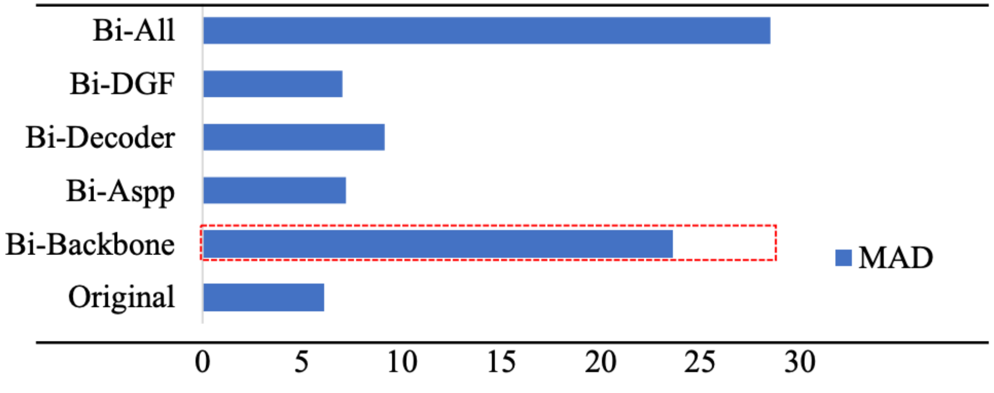
Matting model aims to break down a frame I into a foreground F and a background B, using an α coefficient to represent the linear combination of the two:

$$I = \alpha F + (1 - \alpha)B.$$

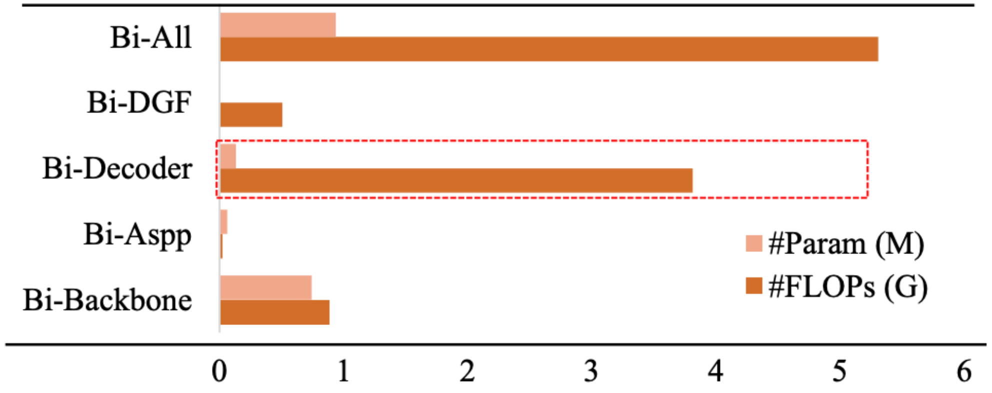


Existing lightweight practice: Robust Video Matting (RVM), MobileNetV3 Encoder + Recurrent Decoder

BiMatting: bottleneck



(a) Accuracy perspective

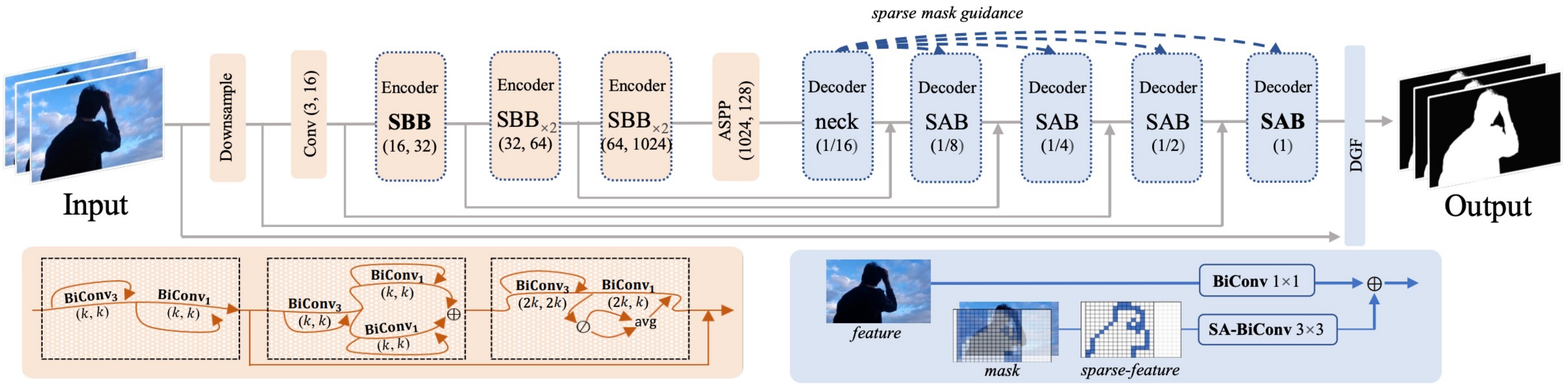


(b) Efficiency perspective

From an accuracy perspective, binarizing the existing lightweight MobileNetV3 backbone in the encoder causes the most significant drop in accuracy among all parts

From an efficiency perspective, the decoder consumes a significant amount of computational resources even after binarization

BiMatting: method



Shrinkable Binarized Block (SBB)

Sparse-Assisted Binarization (SAB)

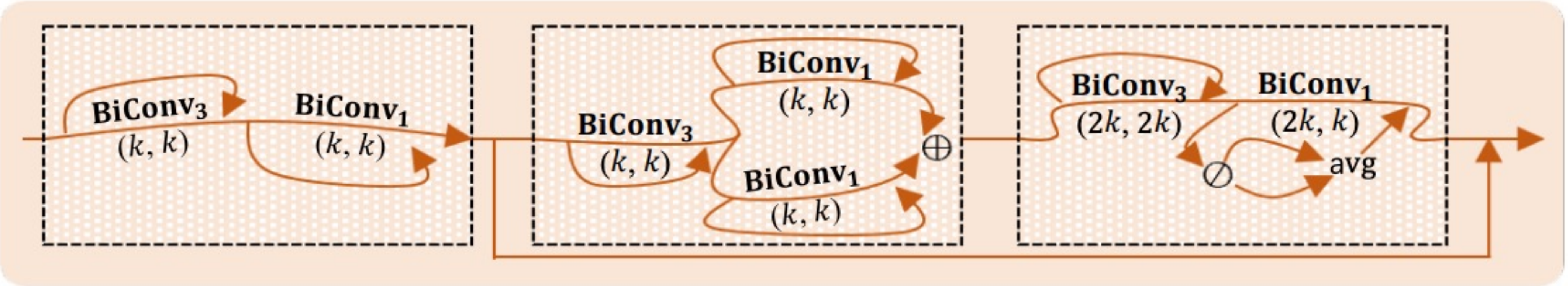
Binarization-evoked Encoder Degradation

$$\textit{MBV3 Block (1):}\quad \boldsymbol{o} = \mathrm{BiConv}_1^{\mathrm{eq}}(\mathrm{GBiConv}_n^{\mathrm{eq}}(\mathrm{BiConv}_1^{\mathrm{eq}}(\boldsymbol{x}))) + \boldsymbol{x}, \quad s.t.\ c^{\boldsymbol{x}} = c^{\boldsymbol{o}}$$

$$\textit{MBV3 Block (2):}\quad \boldsymbol{o} = \mathrm{BiConv}_1^{\mathrm{dn}}(\mathrm{GBiConv}_n^{\mathrm{eq}}(\mathrm{BiConv}_1^{\mathrm{up}}(\boldsymbol{x}))) + [c^{\boldsymbol{x}} = c^{\boldsymbol{o}}]\boldsymbol{x},$$

BiMatting: method

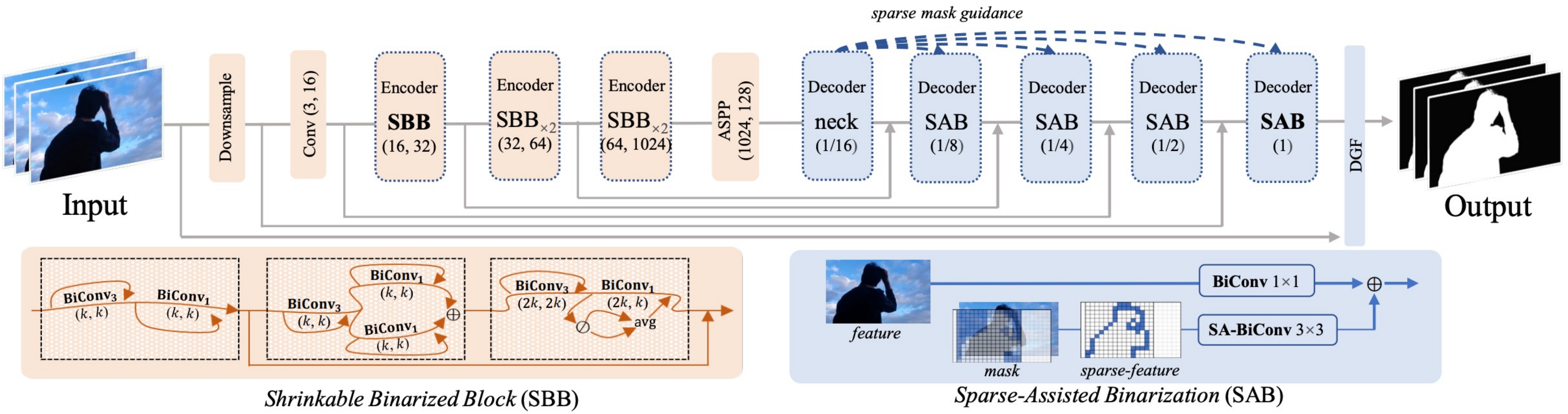
Shrinkable Binarized Block for Accurate Encoder: the crucial paradigm of an accurate binarized encoder is the computation-dense form of binarized block.



Shrinkable Binarized Block (SBB)

$$\textbf{\textit{SBB}}: \quad \boldsymbol{o} = \theta^{\mathrm{dn}} \cdot \theta^{\mathrm{up}}(\boldsymbol{x}') + \boldsymbol{x}', \quad \boldsymbol{x}' = \theta^{\mathrm{eq}}(\boldsymbol{x})[c^{\boldsymbol{x}} = c^{\boldsymbol{o}}] + \theta^{\mathrm{up}}(\boldsymbol{x})\left[c^{\boldsymbol{x}} = \frac{1}{2}c^{\boldsymbol{o}}\right].$$

BiMatting: method

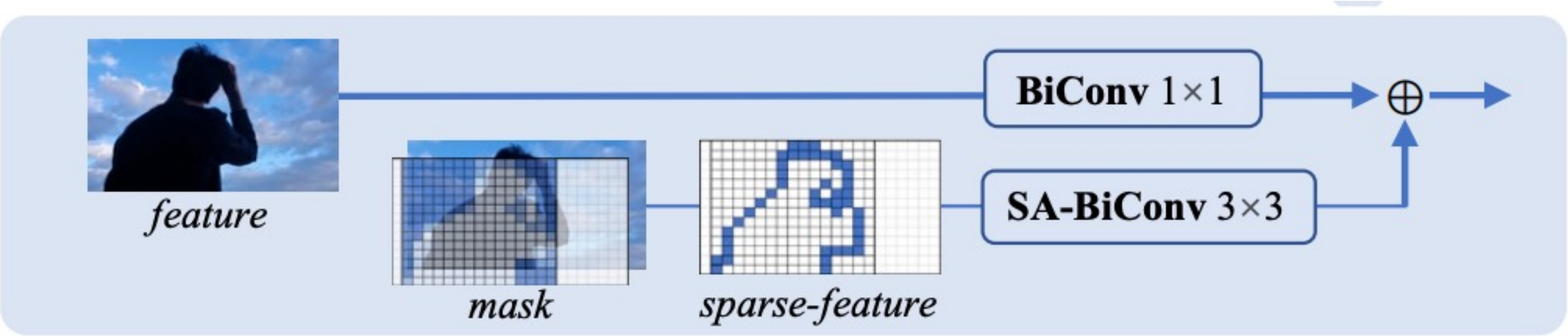


Computational Decoder Redundancy

The computation of this single block in the decoder (the last one in 5 decoder blocks) is even equivalent to 103% of the entire encoder in a binarized baseline.

BiMatting: method

Sparse-Assisted Binarization for Efficient Decoder:



Sparse-Assisted Binarization (SAB)

$$\boldsymbol{SAB}: \quad \boldsymbol{o} = \text{SA-BiConv}_3(\boldsymbol{x}; \text{bilinear}^k(M_{\text{inc}})) + \text{BiConv}_1(\boldsymbol{x}),$$

BiMatting: quantitative results

Table 2: Low-resolution comparison on VM, D646, and AIM datasets. **Bold** indicates the best performance among binarized video matting models and † indicates the results is crashed.

					Alpha					FG
Dataset	Method	#Bit	#FLOPs(G)	#Param(MB)	MAD	MSE	Grad	Conn	dtSSD	MSE
VM 512×288	DeepLabV3	32	136.06	223.66	14.47	9.67	8.55	1.69	5.18	-
	BGMv2	32	8.46	19.4	25.19	19.63	2.28	3.26	2.74	-
	RVM (oracle)	32	4.57	14.5	6.08	1.47	0.88	0.41	1.36	-
	RVM-BNN†	1	0.50	0.57	189.13	184.33	15.01	27.39	3.65	-
	RVM-DoReFa	1	0.52	0.57	51.64	34.50	8.85	7.14	4.09	-
	RVM-ReCU†	1	0.52	0.64	189.13	184.33	15.01	27.39	3.65	-
	RVM-ReAct	1	0.55	0.64	28.49	18.16	6.80	3.74	3.64	-
	BiMatting (Ours)	1	**0.37**	0.67	**12.82**	**6.65**	**2.97**	**1.42**	**2.69**	-
D646 512×512	DeepLabV3	32	241.89	223.66	24.50	20.1	20.30	6.41	4.51	-
	BGMv2	32	16.48	19.4	43.62	38.84	5.41	11.32	3.08	2.60
	RVM (oracle)	32	8.12	14.5	7.28	3.01	2.81	1.83	1.01	2.93
	RVM-BNN†	1	0.88	0.57	281.20	276.85	25.26	73.59	1.08	6.95
	RVM-DoReFa	1	0.92	0.57	133.63	116.69	17.09	35.08	2.58	6.97
	RVM-ReCU†	1	0.92	0.64	281.20	276.85	25.26	73.59	1.08	6.95
	RVM-ReAct	1	0.97	0.64	56.41	43.10	14.05	14.85	2.56	6.85
	BiMatting (Ours)	1	**0.66**	0.67	**32.74**	**24.48**	**9.34**	**8.62**	**2.21**	**5.86**
AIM 512×512	DeepLabV3	32	241.89	223.66	29.64	23.78	20.17	7.71	4.32	-
	BGMv2	32	16.48	19.4	44.61	39.08	5.54	11.60	2.69	3.31
	RVM (oracle)	32	8.12	14.5	14.84	8.93	4.35	3.83	1.01	5.01
	RVM-BNN†	1	0.88	0.57	327.02	321.15	23.80	85.55	0.75	7.84
	RVM-DoReFa	1	0.92	0.57	129.29	107.79	17.31	34.18	2.62	7.85
	RVM-ReCU†	1	0.92	0.64	327.02	321.15	23.80	85.55	0.75	7.84
	RVM-ReAct	1	0.97	0.64	59.90	44.08	14.32	15.90	2.37	8.00
	BiMatting (Ours)	1	**0.66**	0.67	**35.17**	**26.53**	**9.42**	**9.24**	**1.82**	**7.00**

BiMatting: quantitative results

NEURAL INFORMATION PROCESSING SYSTEMS

Table 3: High-resolution comparison on VM, D646, and AIM datasets. * indicates using the officially released model directly [40].

Dataset	Method	#Bit	#FLOPs(G)	#Param(MB)	SAD	MSE	Grad	dtSSD
VM	RVM	32	4.15	14.5	6.57	1.93	10.55	1.90
1920×1080	BGMv2*	32	9.86	19.4	49.83	44.71	74.71	4.09
	RVM-ReAct	1	0.53	0.64	31.60	20.29	34.28	4.08
	BiMatting (Ours)	1	**0.38**	0.67	**18.16**	**11.15**	**21.90**	**3.25**
D646	RVM	32	8.37	14.5	8.67	4.28	30.06	1.64
2048×2048	BGMv2*	32	15.19	19.4	57.40	52.00	149.20	2.56
	RVM-ReAct	1	1.07	0.64	57.38	42.14	71.24	3.03
	BiMatting (Ours)	1	**0.77**	0.67	**52.85**	**44.08**	**61.60**	3.12
AIM	RVM	32	8.37	14.5	14.89	9.01	34.97	1.71
2048×2048	BGMv2*	32	15.19	19.4	45.76	38.75	124.06	2.02
	RVM-ReAct	1	1.07	0.64	57.38	42.14	71.24	3.03
	BiMatting (Ours)	1	**0.77**	0.67	**48.27**	**38.37**	**61.72**	**2.80**

BiMatting: visual results

Binarization for Efficient Computer Vision

2. Binarization (1 bit)

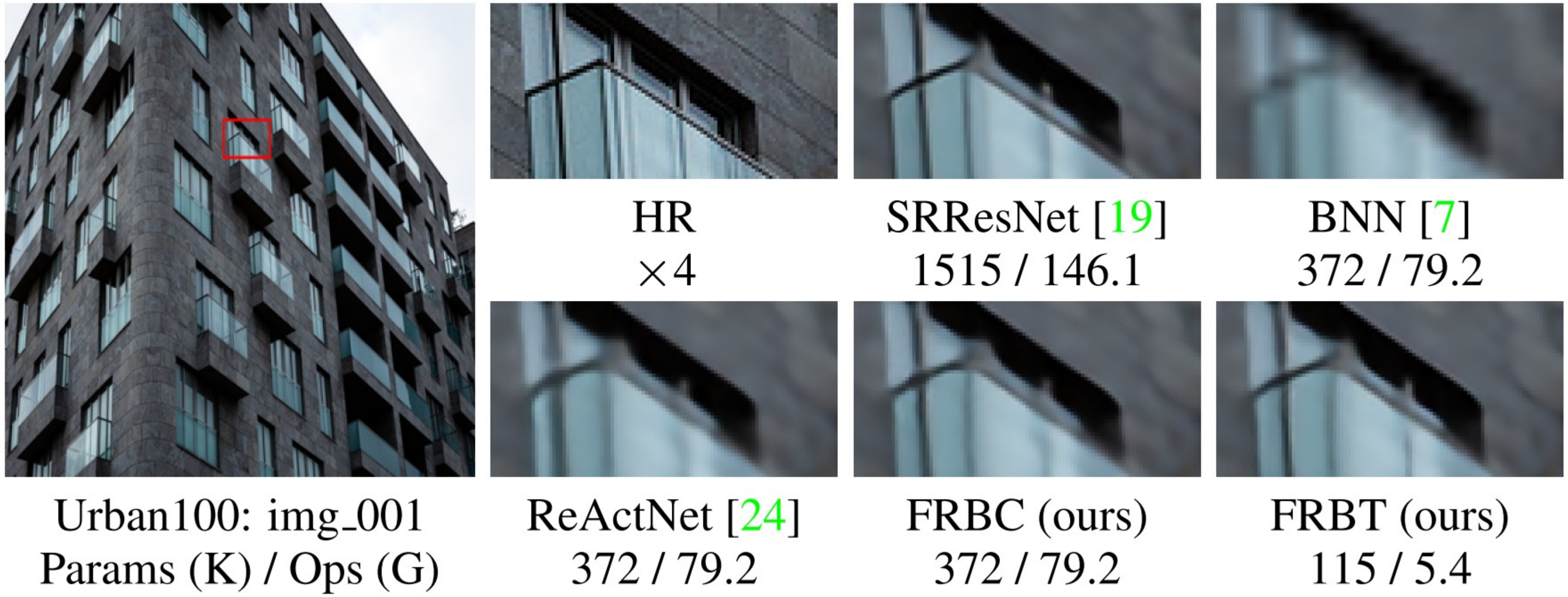
   2.1. BiBench: Benchmarking and Analyzing Network Binarization

   2.2. BiMatting: Efficient Video Matting via Binarization

   2.3. Flexible Residual Binarization for Image Super-Resolution

Binarization for Efficient Computer Vision

Flexible Residual Binarization for Image Super-Resolution



ICLR 2024
Submission

Visual samples of image SR (×4). Our FRBC and FRBT achieves better visual reconstruction. We set the input size as 3×320×180 for Ops calculation.

FRB: motivation

- Why residual binarization?

--**The weights are binarized from full-precision (i.e., 32-bit) to 1-bit, being hard to extract high-frequency information.**

--**Binarizing activations (i.e., features) would directly lose high-frequency information, which is the key component that SR networks try to recover.**

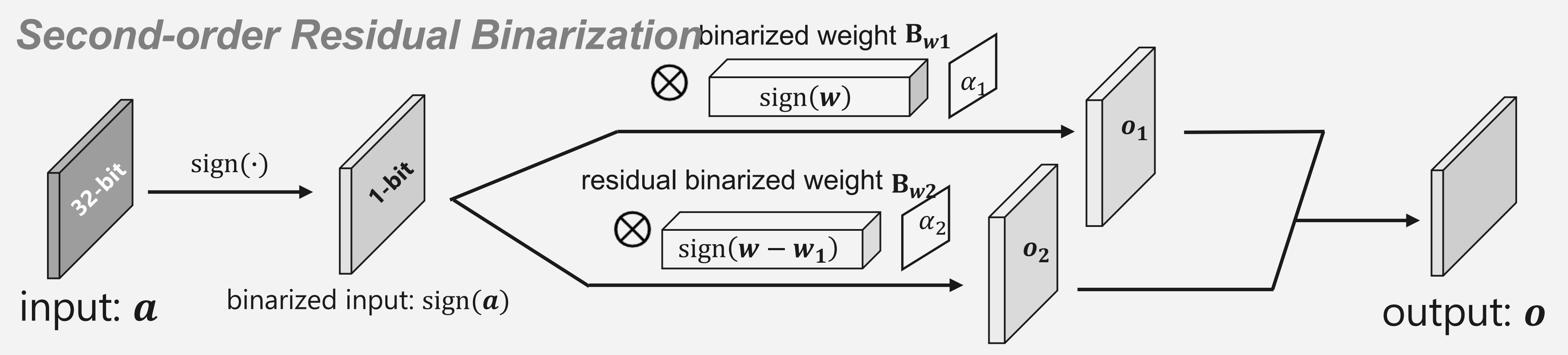
- Why distillation-guided binarization training?

--**After the computation operations between binarized weights and activations, the output would further lose pixel-wise information with high uncertainty.**

FRB: contribution

- We propose Flexible Residual Binarization (FRB) to accurately binarize full-precision SR networks.
- We propose an effective **Second-order Residual Binarization** (SRB), which binarizes the SR network with its weight residuals.
- We propose **Distillation-guided Binarization Training** (DBT), which transfers full-precision knowledge to the binarized model.
- We employ our **FRB to binarize CNN and Transformer** based SR networks respectively, resulting in two binarized baselines: FRBC and FRBT.

FRB: method



First and second order binarization can be expressed as

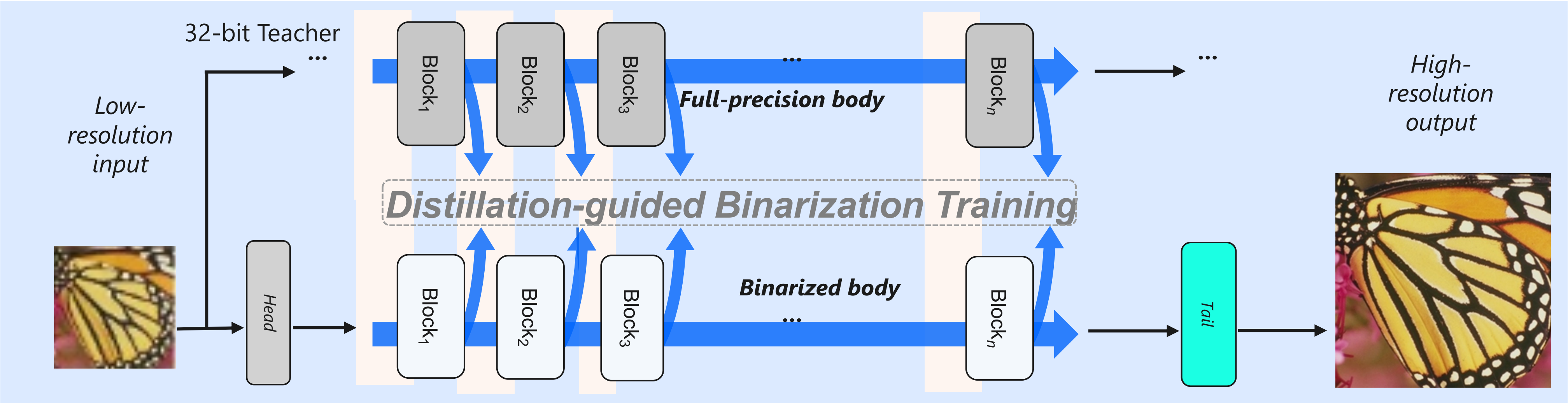
$$\mathbf{B}_{\boldsymbol{w}\mathbf{1}} = \alpha_1 \operatorname{sign}(\boldsymbol{w}), \qquad \alpha_1 = \frac{1}{n}\|\boldsymbol{w}\|_1,$$

$$\mathbf{B}_{\boldsymbol{w}\mathbf{2}} = \alpha_2 \operatorname{sign}(\boldsymbol{w} - \mathbf{B}_{\boldsymbol{w}\mathbf{1}}), \; \alpha_2 = \frac{1}{n}\|\boldsymbol{w} - \mathbf{B}_{\boldsymbol{w}\mathbf{1}}\|_1.$$

Output of binarization

$$o = \operatorname{sign}(\boldsymbol{a}) \otimes \mathbf{B}_{\boldsymbol{w}\mathbf{1}} + \operatorname{sign}(\boldsymbol{a}) \otimes \mathbf{B}_{\boldsymbol{w}\mathbf{2}}$$

FRB: method



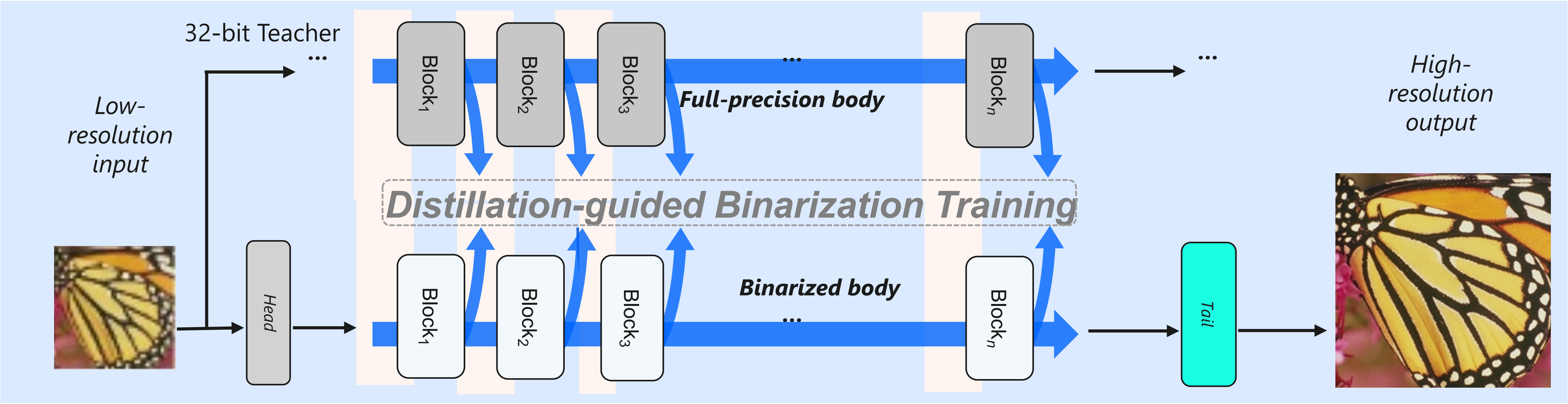
Reformulate SR pipeline as follows

$$I_{SR} = \mathcal{F}_{\mathrm{BSR}}(I_{\mathrm{LR}}; \mathbf{\Theta}) = \prod_{i=1}^{n} Blk_{\mathrm{BSR}_i}(I_{\mathrm{LR}}; \mathbf{\Theta})$$

Distortion caused by binarization, a.k.a. difference between full precision and binarization

$$\mathcal{D}_k = \prod_{i=1}^{k} Blk_{\mathrm{SR}_i}(I_{\mathrm{LR}}; \mathbf{\Theta}) - \prod_{i=1}^{k} Blk_{\mathrm{BSR}_i}(I_{\mathrm{LR}}; \mathbf{\Theta})$$

FRB: method



Normalized representation

$$R_{\mathrm{BSR}_k} = \frac{\left(\prod_{i=1}^{k} Blk_{\mathrm{BSR}_i}(I_{\mathrm{LR}};\boldsymbol{\Theta})\right)^2}{\left\|\left(\prod_{i=1}^{k} Blk_{\mathrm{BSR}_i}(I_{\mathrm{LR}};\boldsymbol{\Theta})\right)^2\right\|_{\ell 2}}$$

Distillation-guided binarization training loss

$$\min \mathcal{L}_{\mathrm{DBT}} = \sum_{i=1}^{n} \hat{\mathcal{D}}_i = \sum_{i=1}^{n} \|R_{\mathrm{SR}_i} - R_{\mathrm{BSR}_i}\|_{\ell 2}$$

FRB: quantitative results

Method	Scale	Bits (W/A)	Set5		Set14		B100		Urban100		Manga109	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	×2	-/-	33.66	0.9299	30.24	0.8688	29.56	0.8431	26.88	0.8403	30.80	0.9339
SRResNet [19]	×2	32/32	38.00	0.9605	33.59	0.9171	32.19	0.8997	32.11	0.9282	38.56	0.9770
BNN [7]	×2	1/1	32.25	0.9118	29.25	0.8406	28.68	0.8104	25.96	0.8088	29.16	0.9127
DoReFa [46]	×2	1/1	36.76	0.9550	32.44	0.9072	31.31	0.8883	29.26	0.8945	35.81	0.9682
Bi-Real [25]	×2	1/1	32.32	0.9123	29.47	0.8424	28.74	0.8111	26.35	0.8161	29.64	0.9167
IRNet [31]	×2	1/1	37.27	0.9579	32.92	0.9115	31.76	0.8941	30.63	0.9122	36.77	0.9724
BAM [40]	×2	1/1	37.21	0.9560	32.74	0.9100	31.60	0.8910	30.20	0.9060	N/A	N/A
BTM [16]	×2	1/1	37.22	0.9575	32.93	0.9118	31.77	0.8945	30.79	0.9146	36.76	0.9724
ReActNet [24]	×2	1/1	37.26	0.9579	32.97	0.9124	31.81	0.8954	30.85	0.9156	36.92	0.9728
FRBC (ours)	×2	1/1	37.63	0.9590	33.14	0.9137	31.89	0.8956	31.00	0.9164	37.77	0.9749
FRBC+ (ours)	×2	1/1	37.78	0.9595	33.23	0.9145	31.97	0.8965	31.13	0.9178	38.10	0.9758
Bicubic	×4	-/-	28.42	0.8104	26.00	0.7027	25.96	0.6675	23.14	0.6577	24.89	0.7866
SRResNet [19]	×4	32/32	32.16	0.8951	28.60	0.7822	27.58	0.7364	26.11	0.7870	30.46	0.9089
BNN [7]	×4	1/1	27.56	0.7896	25.51	0.6820	25.54	0.6466	22.68	0.6352	24.19	0.7670
DoReFa [46]	×4	1/1	30.33	0.8601	27.40	0.7526	26.83	0.7104	24.29	0.7175	27.00	0.8470
Bi-Real [25]	×4	1/1	27.75	0.7935	25.79	0.6879	25.59	0.6478	22.91	0.6450	24.57	0.7752
IRNet [31]	×4	1/1	31.38	0.8835	28.08	0.7679	27.24	0.7227	25.21	0.7536	28.97	0.8863
BAM [40]	×4	1/1	31.24	0.8780	27.97	0.7650	27.15	0.7190	24.95	0.7450	N/A	N/A
BTM [16]	×4	1/1	31.43	0.8850	28.16	0.7706	27.29	0.7256	25.34	0.7605	29.19	0.8912
ReActNet [24]	×4	1/1	31.54	0.8859	28.19	0.7705	27.31	0.7252	25.35	0.7603	29.25	0.8912
FRBC (ours)	×4	1/1	31.68	0.8881	28.29	0.7739	27.36	0.7279	25.49	0.7644	29.51	0.8962
FRBC+ (ours)	×4	1/1	31.82	0.8902	28.38	0.7759	27.42	0.7293	25.58	0.7668	29.72	0.8988

Table 2: Quantitative results in CNN based binarized image SR methods. SRResNet is used as the full-precision backbone. Bits (W/A) denote the bits of weights and activations. The best and second best results are colored with red and cyan.

FRB: quantitative results

Method	Scale	Bits (W/A)	Set5		Set14		B100		Urban100		Manga109	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SwinIR_S [22]	×2	32/32	38.14	0.9611	33.86	0.9206	32.31	0.9012	32.76	0.9340	39.12	0.9783
FRBT (ours)	×2	1/1	37.62	0.9591	33.19	0.9143	31.93	0.8966	31.02	0.9173	37.78	0.9751
FRBT+ (ours)	×2	1/1	37.76	0.9596	33.27	0.9152	32.00	0.8974	31.17	0.9187	38.12	0.9759
SwinIR_S [22]	×4	32/32	32.44	0.8976	28.77	0.7858	27.69	0.7406	26.47	0.7980	30.92	0.9151
FRBT (ours)	×4	1/1	31.71	0.8883	28.30	0.7742	27.38	0.7291	25.47	0.7650	29.52	0.8964
FRBT+ (ours)	×4	1/1	31.86	0.8903	28.39	0.7761	27.43	0.7305	25.58	0.7674	29.78	0.8996

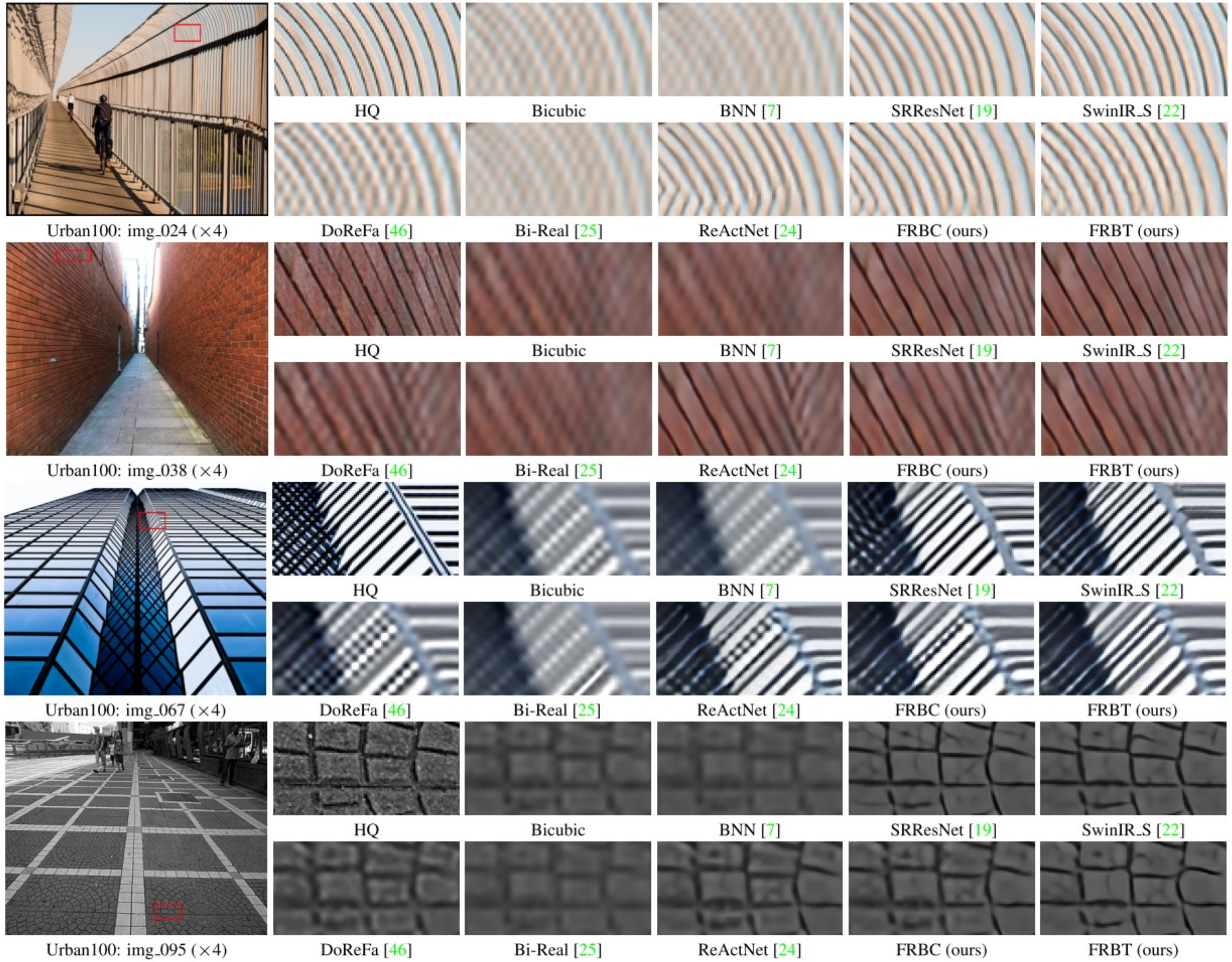
Table 3: Quantitative results in Transformer based binarized image SR methods. We use SwinIR_S as the backbone. We find quantization of Transformer models cause a significant quality loss. This is an interesting problem for future work.

FRB: model complexity

Method	Bits (W/A)	Params (K) (↓ Compr. Ratio)	Ops (G) (↓ Compr. Ratio)	Urban100	
				PSNR	SSIM
SRResNet	32 / 32	1367 (0%)	85.4 (0%)	32.11	0.9282
FRBC (ours)	1 / 1	225 (↓ 83.5%)	18.6 (↓ 78.2%)	31.00	0.9164
SwinIR_S	32 / 32	910 (0%)	62.4 (0%)	32.76	0.9340
FRBT (ours)	1 / 1	95 (↓ 89.6%)	4.3 (↓ 93.1%)	31.02	0.9173

Table 4: Compression ratio of SRResNet and SwinIR_S ($\times$2). Bits (W/A) denote the weights and activations bit number. We set the input size as 3$\times$320$\times$180 for Ops calculation. Our Transformer baseline FRBT performs better than CNN one FRBC with a larger compression ratio.

FRB: visual results



Quantization for Efficient Computer Vision

3. Quantization (2-8 bit)

3.1. QuantSR: Accurate Low-bit Quantization for Efficient Image Super-Resolution

Quantization for Efficient Computer Vision

QuantSR: Accurate Low-bit Quantization for Efficient Image Super-Resolution

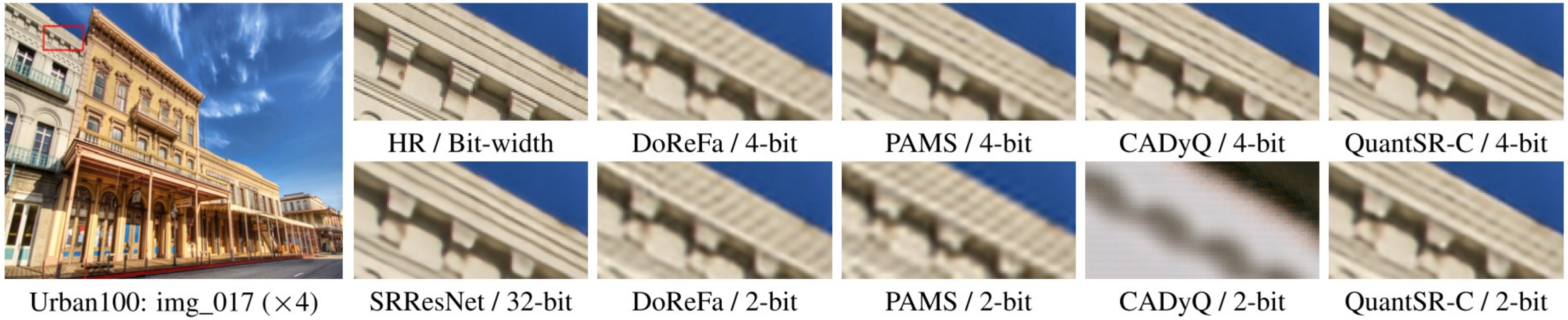


Figure 1: Visual comparison (×4) with quantized lightweight SR models in terms of 4-bit and 2-bit. We use SRResNet [21] as the full-precision SR backbone and quantize it with low bit width. We compare our QuantSR-C with recent quantization methods (*i.e.*, DoReFa [44], PAMS [23], and CADyQ [11]). Our QuantSR-C performs obviously better than others in both 4-bit and 2-bit cases.

QuantSR: motivation

- Narrow performance gap between full-precision and quantized ones

--**significant performance degradation, particularly when using ultra-low bit width, e.g., 2-4 bits**.

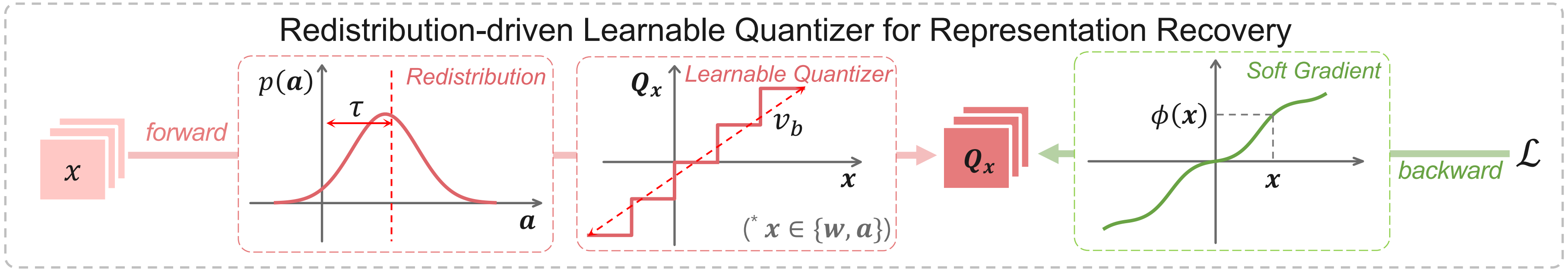
- Different from 1-bit quantization

--**1-bit quantization suffers from a much larger performance gap and has a different hardware implementation in practice when compared with low-bit quantization settings.**

QuantSR: contribution

- We propose QuantSR, a novel accurate quantization scheme for efficient image SR.
- We propose a **Redistribution-driven Learnable Quantizer** (RLQ). Specifically, our RLQ diversifies quantized representation and gradient information by redistribution in quantizers.
- We propose a **Depth-dynamic Quantized Architecture** (DQA) to achieve better performance with the same network depth.
- We employ our QuantSR to compress CNN- and Transformer- based SR networks to lower bit-width, resulting in the corresponding quantized baselines, QuantSR-C and QuantSR-T.

QuantSR: method



Basic quantization framework: forward.

$$Q^b(\boldsymbol{x}) = \text{round}\left(\frac{\text{clip}(\boldsymbol{x})}{v_b}\right) v_b \qquad v_b = \frac{\max(\|\boldsymbol{x}\|_1)}{2^{b-1}-1}$$

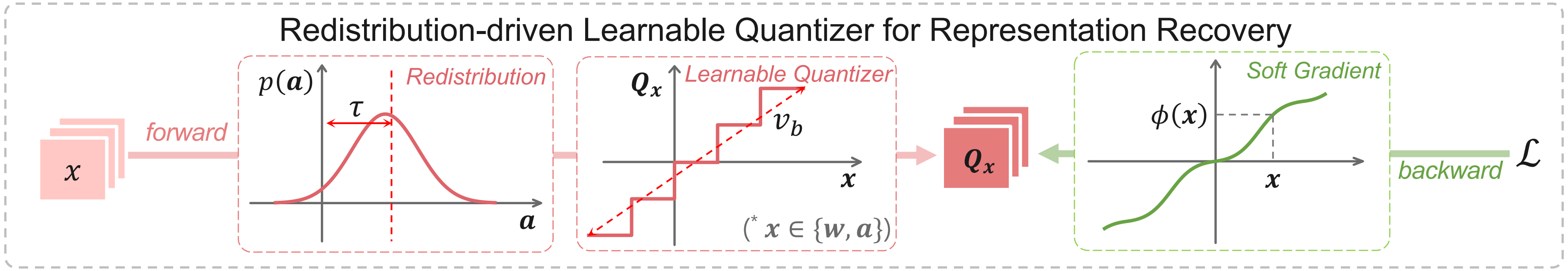
Basic quantization framework: backward. Straight-through estimation (STE) approximate the gradient of parameters

$$\frac{\partial Q^b(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{cases} 1 & \text{if } \boldsymbol{x} \in (-a, a) \\ 0 & \text{otherwise} \end{cases}$$

RLQ can be expressed as

$$Q^b_{\text{RLQ}}(\boldsymbol{x}, \hat{v}_b, \hat{\tau}) = \text{round}\left(\phi\left(\frac{\text{clip}(\boldsymbol{x}+\hat{\tau})}{\hat{v}_b}\right)\right)\hat{v}_b \qquad \phi(\boldsymbol{x}) = \frac{\tanh\left(2(\boldsymbol{x}-\lfloor\boldsymbol{x}\rfloor)-1\right)}{\tanh 1} + \lfloor\boldsymbol{x}\rfloor + 2^{-1}$$

QuantSR: method



RLQ can be expressed as

$$Q^b_{\mathrm{RLQ}}(\boldsymbol{x}, \hat{v}_b, \hat{\tau}) = \mathrm{round}\left(\phi\left(\frac{\mathrm{clip}(\boldsymbol{x} + \hat{\tau})}{\hat{v}_b}\right)\right)\hat{v}_b$$

$$\phi(\boldsymbol{x}) = \frac{\tanh\left(2(\boldsymbol{x} - \lfloor \boldsymbol{x} \rfloor) - 1\right)}{\tanh 1} + \lfloor \boldsymbol{x} \rfloor + 2^{-1}$$

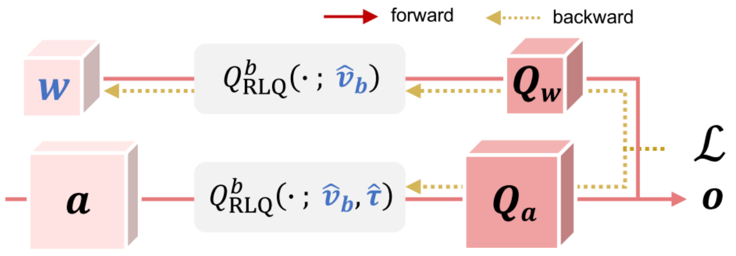


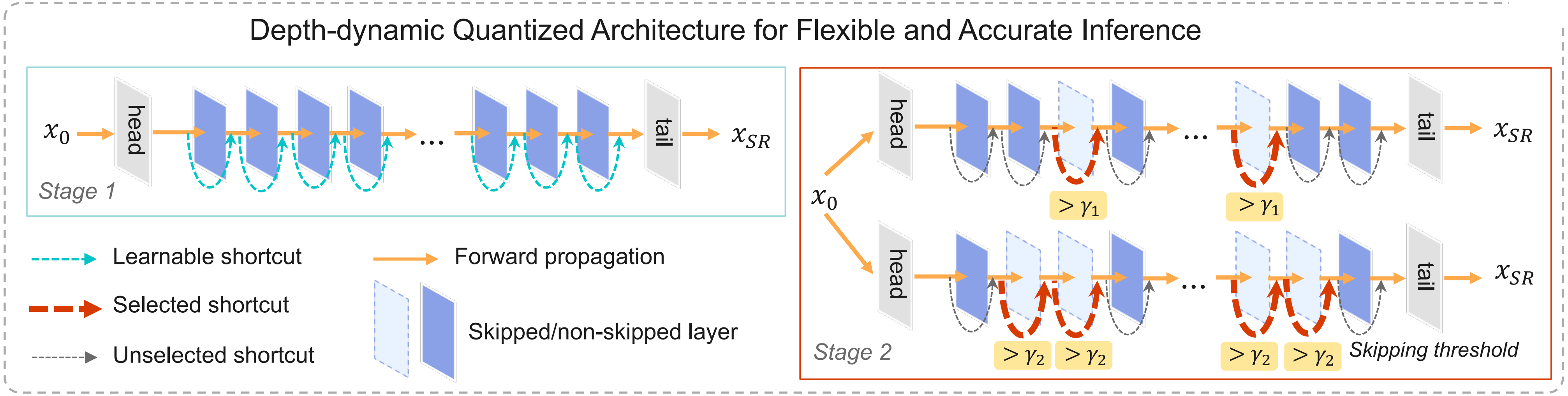
Figure 3: Forward and backward propagation of RLQ. Blue notations are learnable parameters.

The derivative w.r.t. the input and learnable parameters used in the backward pass are

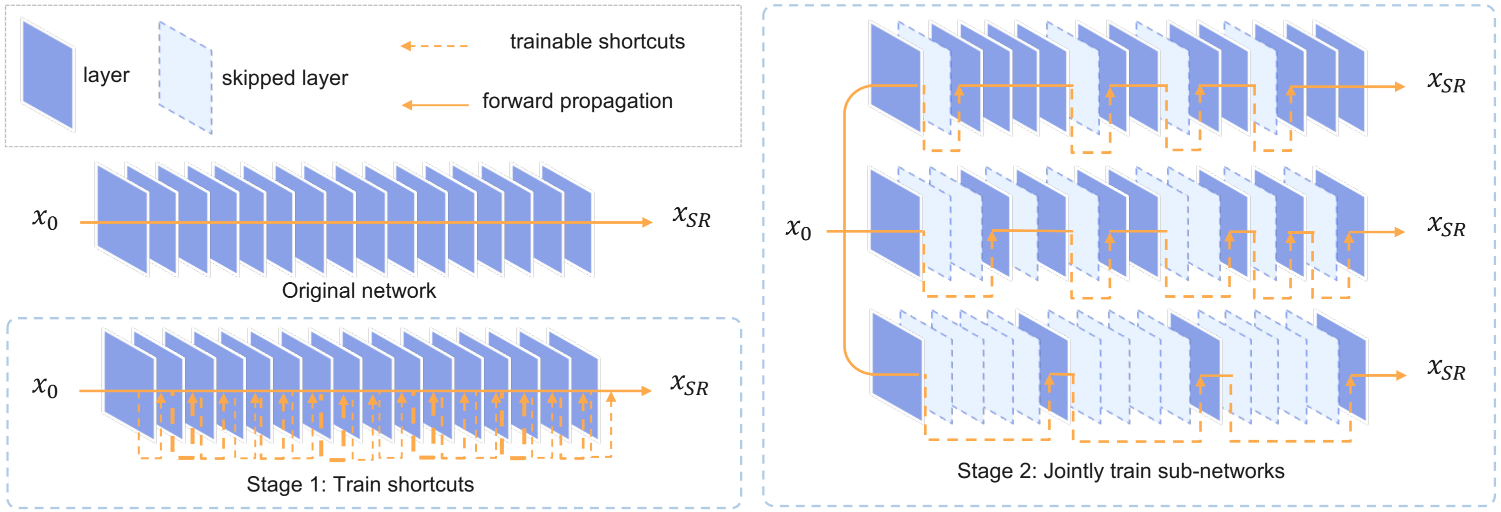
$$\frac{\partial Q^b_{\mathrm{RLQ}}(\boldsymbol{x}, \hat{v}_b, \hat{\tau})}{\partial \boldsymbol{x}} = \begin{cases} \frac{\partial \phi(\boldsymbol{x}+\hat{\tau})}{\partial \boldsymbol{x}} & \text{if } \boldsymbol{x} \in (-a, a) \\ 0 & \text{otherwise} \end{cases}, \qquad \frac{\partial Q^b_{\mathrm{RLQ}}(\boldsymbol{x}, \hat{v}_b, \hat{\tau})}{\partial \hat{\tau}} = 1 + \frac{\partial \phi(\boldsymbol{x} + \hat{\tau})}{\partial \hat{\tau}}$$

$$\frac{\partial Q^b_{\mathrm{RLQ}}(\boldsymbol{x}, \hat{v}_b, \hat{\tau})}{\partial \hat{v}_b} = \begin{cases} \mathrm{round}\left(\frac{\boldsymbol{x}+\hat{\tau}}{\hat{v}_b}\right) + \frac{\partial \phi((x+\hat{\tau})\hat{v}_b^{-1})}{\partial \hat{v}_b} & \text{if } \boldsymbol{x} \in (-a, a) \\ -a \text{ or } a & \text{otherwise} \end{cases}.$$

QuantSR: method



Details



QuantSR: quantitative results

Method	Scale	#Bit	Set5		Set14		B100		Urban100		Manga109	
		(w/a)	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	×2	-/-	33.66	0.9299	30.24	0.8688	29.56	0.8431	26.88	0.8403	30.80	0.9339
SRResNet [21]	×2	32/32	38.00	0.9605	33.59	0.9171	32.19	0.8997	32.11	0.9282	38.56	0.9770
SwinIR_S [26]	×2	32/32	38.14	0.9611	33.86	0.9206	32.31	0.9012	32.76	0.9340	39.12	0.9783
DoReFa [44]	×2	8/8	37.32	0.9520	32.90	0.8680	31.69	0.8504	30.32	0.8800	37.01	0.9450
CADyQ [11]	×2	8/8	37.79	0.9590	33.37	0.9150	32.02	0.8980	31.53	0.9230	38.06	0.9760
DoReFa [44]	×2	4/4	37.31	0.9510	32.48	0.9091	31.64	0.8901	30.18	0.8780	36.95	0.9440
PAMS [23]	×2	4/4	37.67	0.9588	33.19	0.9146	31.90	0.8966	31.10	0.9194	37.62	0.9400
CADyQ [11]	×2	4/4	37.58	0.9580	33.14	0.9140	31.87	0.8960	30.94	0.9170	37.31	0.9740
QuantSR-C (ours)	×2	4/4	37.80	0.9597	33.35	0.9158	32.04	0.8979	31.46	0.9221	38.25	0.9762
QuantSR-T (ours)	×2	4/4	38.10	0.9604	33.65	0.9186	32.21	0.8998	32.20	0.9295	38.85	0.9774
DoReFa [44]	×2	2/2	36.91	0.9470	32.55	0.9071	31.41	0.8868	29.60	0.8740	36.132	0.9410
PAMS [23]	×2	2/2	34.04	0.8270	30.91	0.8751	30.11	0.8592	27.57	0.8400	31.79	0.9110
CADyQ [11]	×2	2/2	19.44	0.5610	18.51	0.4810	19.70	0.4760	17.97	0.4550	17.346	0.5830
QuantSR-C (ours)	×2	2/2	37.57	0.9589	33.09	0.9136	31.84	0.8954	30.77	0.9149	37.60	0.9745
QuantSR-T (ours)	×2	2/2	37.55	0.9587	33.12	0.9143	31.89	0.8958	30.96	0.9172	37.61	0.9745
Bicubic	×4	-/-	28.42	0.8104	26.00	0.7027	25.96	0.6675	23.14	0.6577	24.89	0.7866
SRResNet [21]	×4	32/32	32.16	0.8951	28.60	0.7822	27.58	0.7364	26.11	0.7870	30.46	0.9089
SwinIR_S [26]	×4	32/32	32.44	0.8976	28.77	0.7858	27.69	0.7406	26.47	0.7980	30.92	0.9151
DoReFa [44]	×4	4/4	29.57	0.8369	26.82	0.7352	26.47	0.6971	23.75	0.6898	27.89	0.8634
PAMS [23]	×4	4/4	31.59	0.8851	28.20	0.7725	27.32	0.7220	25.32	0.7624	28.86	0.8805
CADyQ [11]	×4	4/4	31.48	0.8830	28.05	0.7690	27.21	0.7240	25.09	0.7520	28.82	0.8840
QuantSR-C (ours)	×4	4/4	32.00	0.8924	28.50	0.7799	27.52	0.7342	25.88	0.7807	30.15	0.9040
QuantSR-T (ours)	×4	4/4	32.18	0.8941	28.63	0.7822	27.59	0.7367	26.11	0.7871	30.49	0.9087
DoReFa [44]	×4	2/2	30.54	0.8610	27.50	0.7538	26.90	0.7098	24.44	0.7242	27.31	0.8502
PAMS [23]	×4	2/2	29.20	0.8239	26.61	0.7273	26.36	0.6934	23.58	0.6812	25.59	0.8012
CADyQ [11]	×4	2/2	19.67	0.5380	19.30	0.4740	19.80	0.4620	17.97	0.4360	17.30	0.5640
QuantSR-C (ours)	×4	2/2	31.30	0.8819	28.08	0.7694	27.23	0.7246	25.13	0.7537	28.81	0.8844
QuantSR-T (ours)	×4	2/2	31.53	0.8845	28.16	0.7715	27.28	0.7274	25.26	0.7609	29.06	0.8898

Table 2: Quantitative results. SRResNet and SwinIR-S are used as full-precision backbones. 'w/a' denotes the weight/activation bits. The best and second best results are colored with red and cyan.

QuantSR: model complexity

Method	#Bit (w/a)	#Blk	Params (K) (↓ Ratio)	Ops (G) (↓ Ratio)	Urban100	
					PSNR	SSIM
SRResNet	32/32	16	1,367 (0%)	90.1 (0%)	32.16	0.8951
QuantSR-C	4/4	32	451 (↓ 67.0%)	29.9 (↓ 66.9%)	32.17	0.8943
		16	303 (↓ 77.8%)	20.2 (↓ 77.5%)	32.00	0.8924
		8	230 (↓ 83.1%)	15.4 (↓ 82.9%)	31.75	0.8894
QuantSR-C	2/2	32	170 (↓ 87.6%)	11.5 (↓ 87.2%)	31.48	0.8849
		16	161 (↓ 88.2%)	10.9 (↓ 87.9%)	31.30	0.8819
		8	156 (↓ 88.6%)	10.6 (↓ 88.3%)	31.04	0.8771

Table 3: Compression ratio of 2-bit and 4-bit SRResNet ($\times 2$), and their input sizes are $3 \times 256 \times 256$ for calculating Ops.

QuantSR: visual results

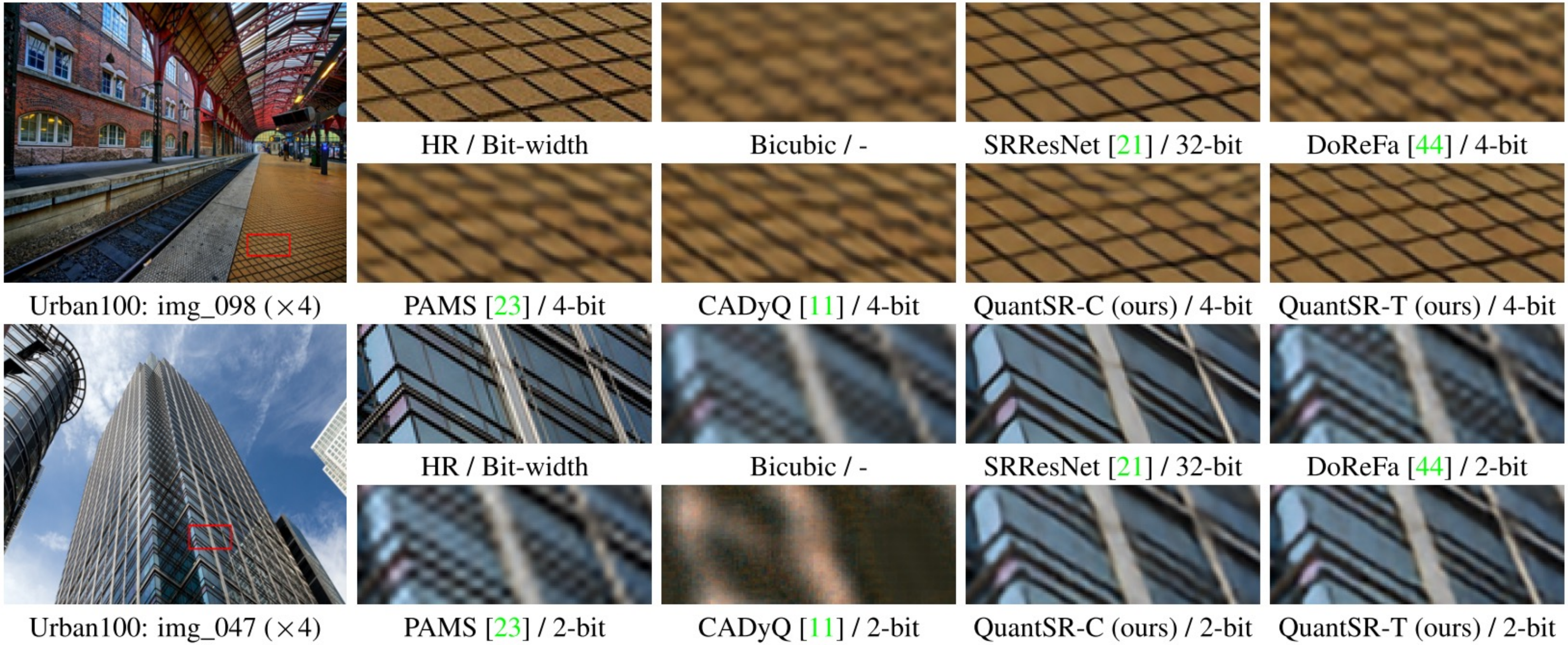


Figure 5: Visual comparison (×4) with lightweight SR in terms of 4-bit and 2-bit.

Summary: next step?

Architecture	Degree	Scale
CNN Quantization	Part Quantization	Quantization of Small Model
↓	↓	↓
Transformer Quantization (SAM, ViT, ...)	Full Quantization	Quantization of Large Model

Thanks!

Haotong Qin
17.10.2023