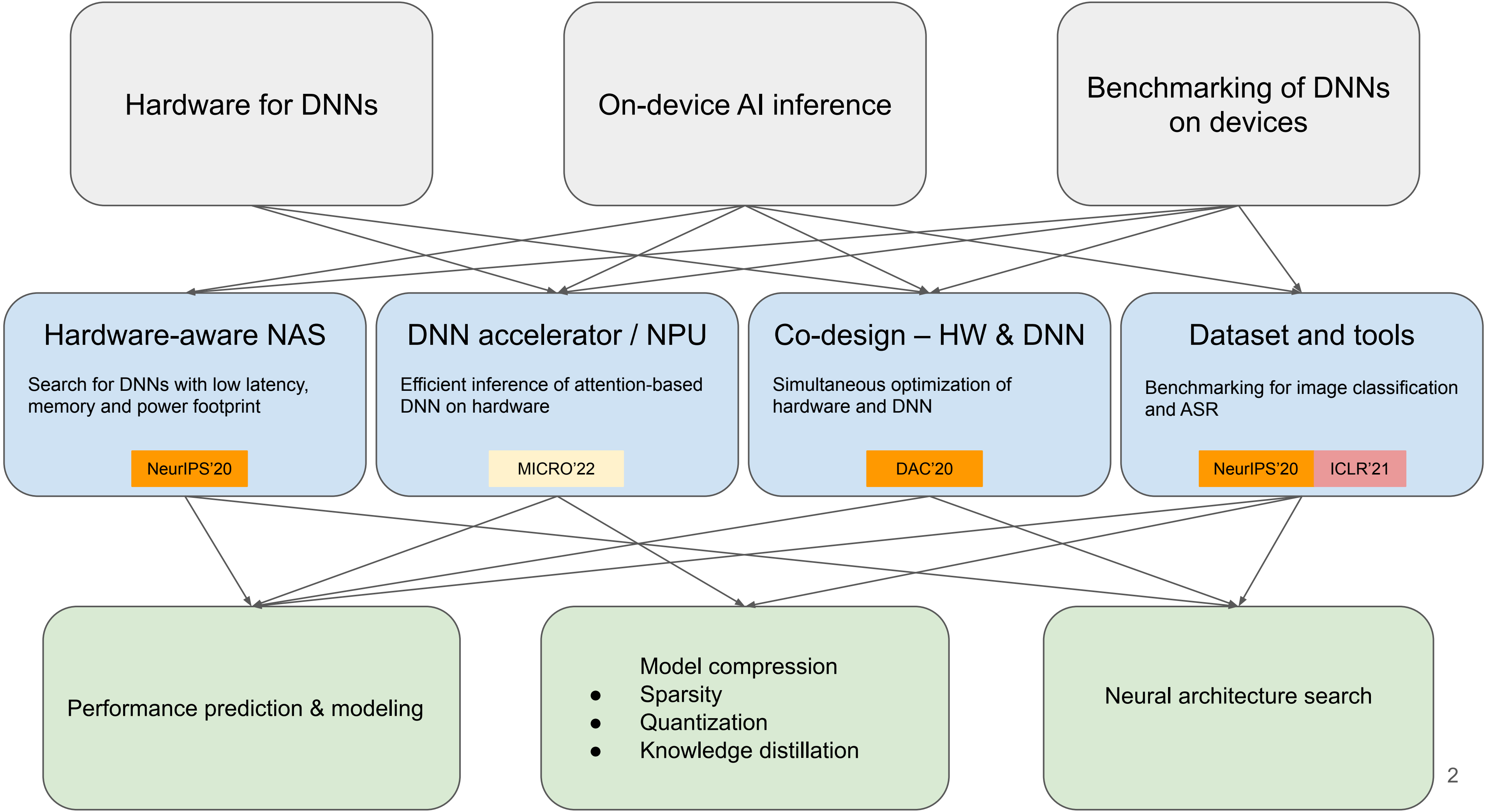
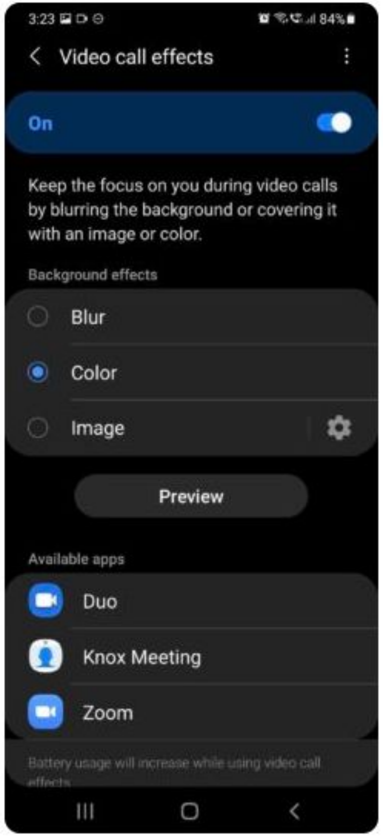
Co-design of Hardware and Neural Network

Thomas Chau


Hardware for DNNs
On-device AI inference
Benchmarking of DNNs on devices
Hardware-aware NAS
Search for DNNs with low latency, memory and power footprint
NeurIPS'20
DNN accelerator / NPU
Efficient inference of attention-based DNN on hardware
MICRO'22
Co-design – HW & DNN
Simultaneous optimization of hardware and DNN
DAC'20
Dataset and tools
Benchmarking for image classification and ASR
NeurIPS'20
ICLR'21
Performance prediction & modeling
Model compression
Sparsity
Quantization
Knowledge distillation
Neural architecture search

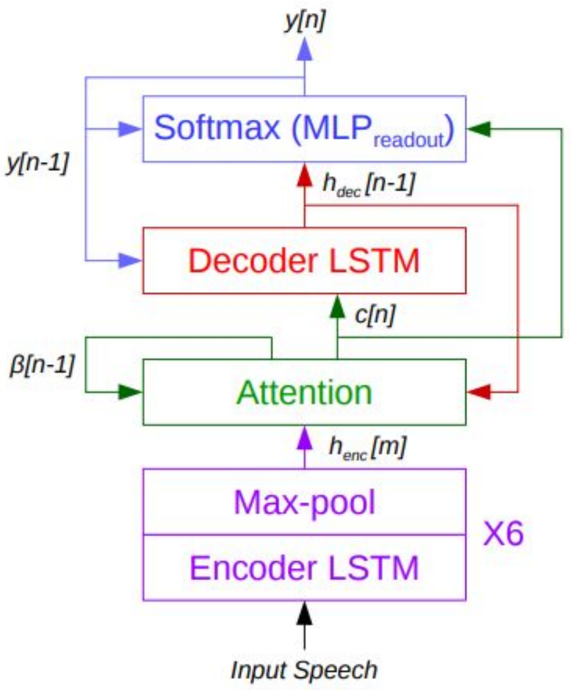
Example application - video segmentation

- Human Semantic Segmentation model used in Google Duo, MS teams, Zoom meeting etc
- Finding light-weight model that meet real-time requirement, without accuracy loss

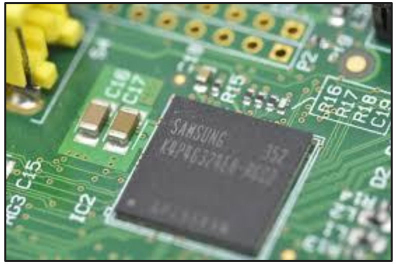


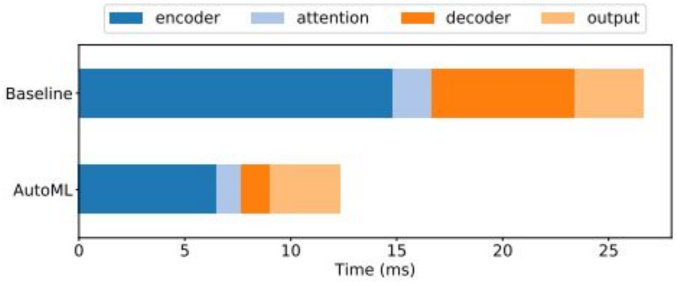
Example application - ASR

- Audio interface of phone, TV, AC, microwaves, …
- Finding highly-compressed ASR models
  - Optimise the per-layer compression ratios
  - Use singular value decomposition (SVD) low-rank matrix factorization as the compression method
  - Search for ranks that have minimal impact on performance



$$m\left\{\overbrace{\begin{bmatrix} \quad A \quad \end{bmatrix}}^{n}\right. \approx m\left\{\overbrace{\begin{bmatrix} X \end{bmatrix}}^{k}\right.\overbrace{\begin{bmatrix} \quad Y \quad \end{bmatrix}}^{n}\}k$$





Runtime on Qualcomm Snapdragon 845 chipset.

Metaverse

Computer vision
E.g. OCR, upscaling

Speech
E.g speech recognition, text to speech

Model size
memory & storage space

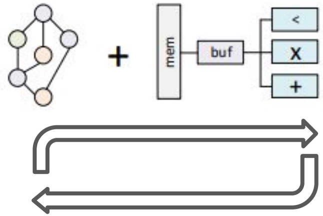
Energy consumption
battery life

Latency and accuracy
user perception

Different models and hardwares to meet different application requirements

Model

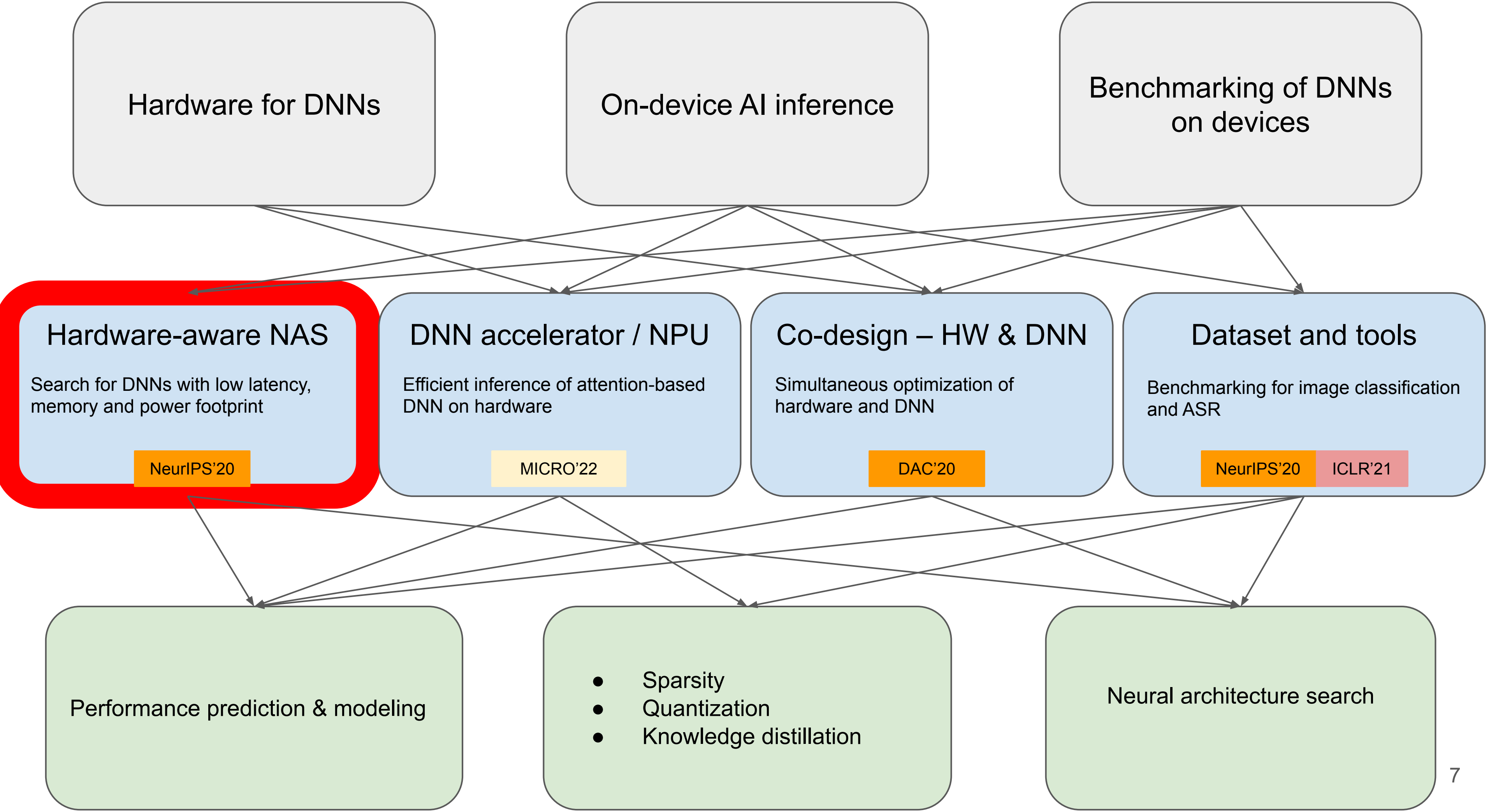
Device



Accuracy | Latency

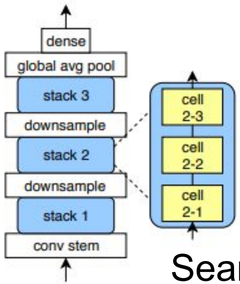
Memory | Energy

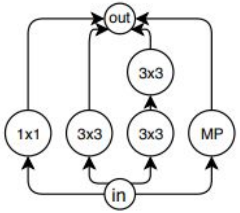
Hardware-aware Neural Architecture Search


Hardware for DNNs
On-device AI inference
Benchmarking of DNNs on devices
Hardware-aware NAS
Search for DNNs with low latency, memory and power footprint
NeurIPS'20
DNN accelerator / NPU
Efficient inference of attention-based DNN on hardware
MICRO'22
Co-design – HW & DNN
Simultaneous optimization of hardware and DNN
DAC'20
Dataset and tools
Benchmarking for image classification and ASR
NeurIPS'20
ICLR'21
Performance prediction & modeling
Sparsity
Quantization
Knowledge distillation
Neural architecture search

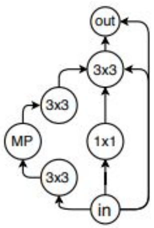
Neural architecture search (NAS)

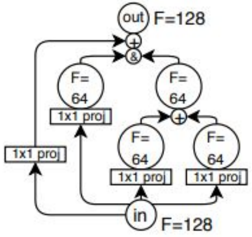
- Automating the design of DNNs
- Search space, search strategy, performance evaluation
- Traditional NAS focus on improving accuracy of DNNs

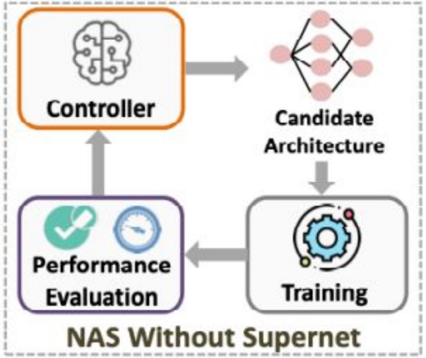


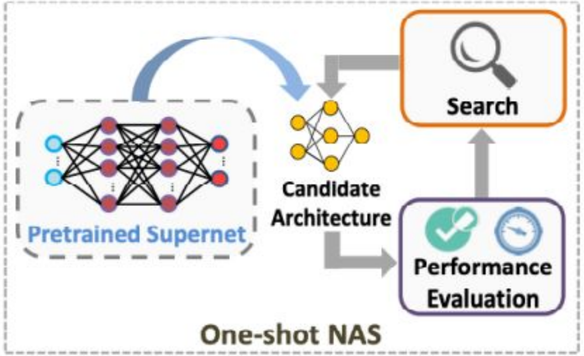


Search space





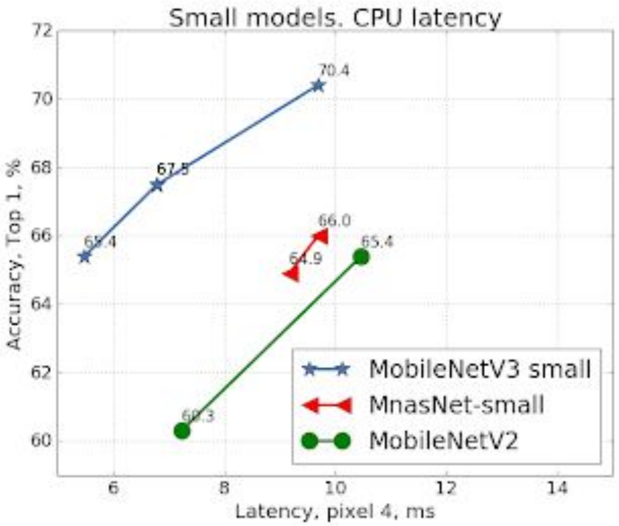


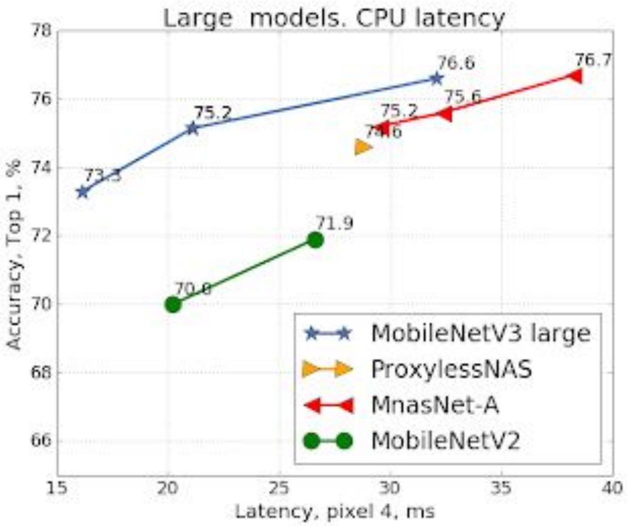


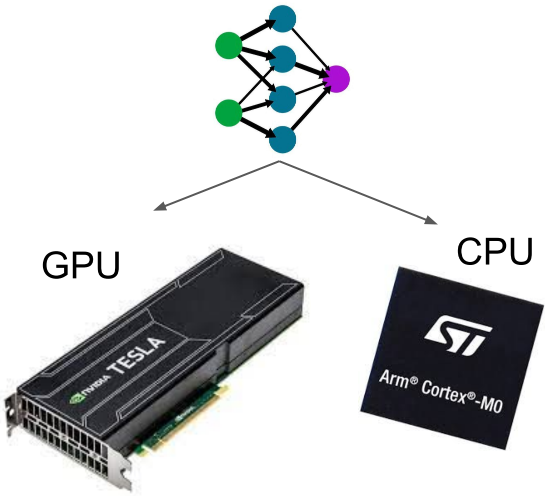
“Hardware-aware” NAS

$$\max_{\mathbf{x}\in\mathcal{X}} \max_{\omega_\mathbf{x}} \; accuracy(\mathbf{x}, \omega_\mathbf{x})$$
$$s.t., \;\; latency(\mathbf{x}; \mathbf{d}) \leq \overline{L}_\mathbf{d}$$

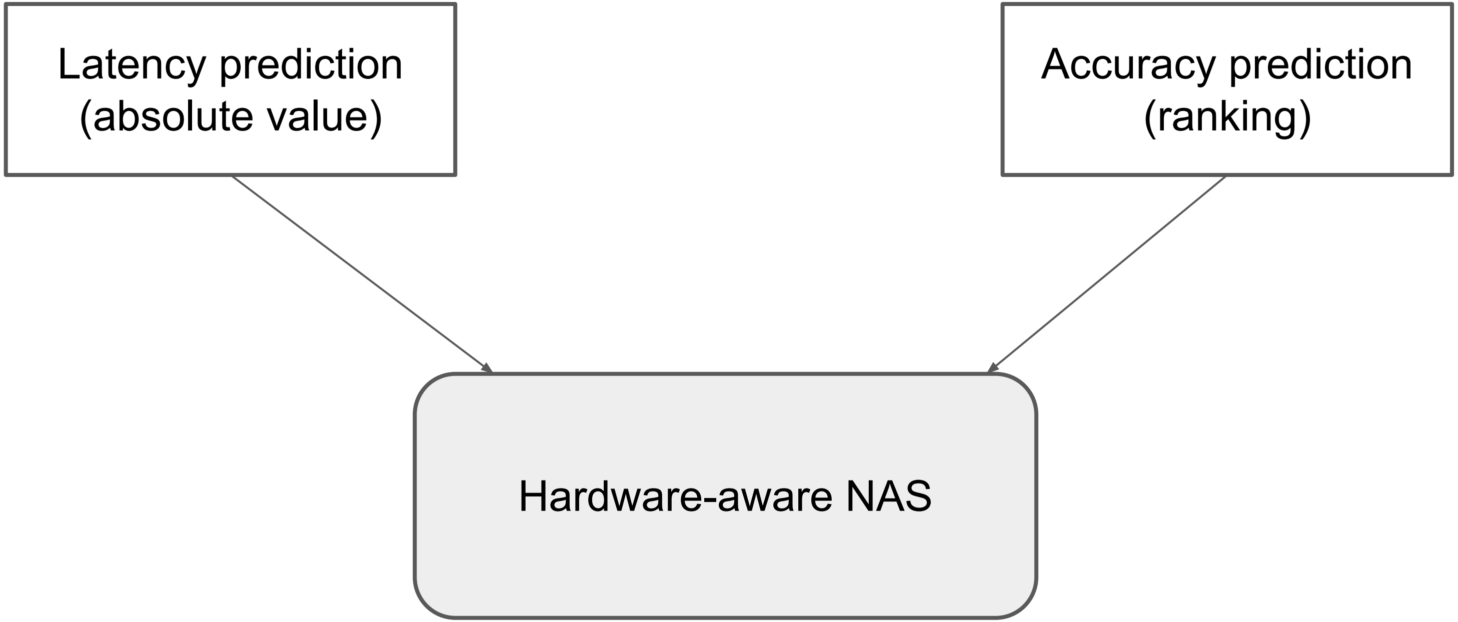
On-device inference	
Performance metrics	Accuracy + Latency, energy, memory
Search space	DNN (+ Hardware)



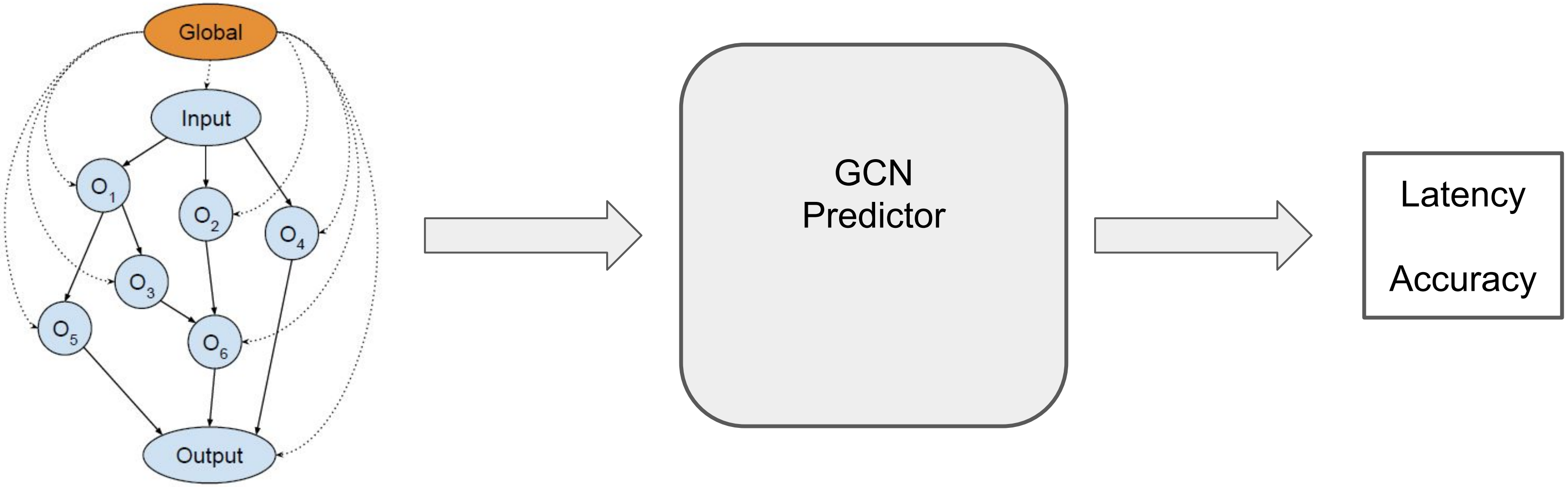




Efficient hardware-aware NAS

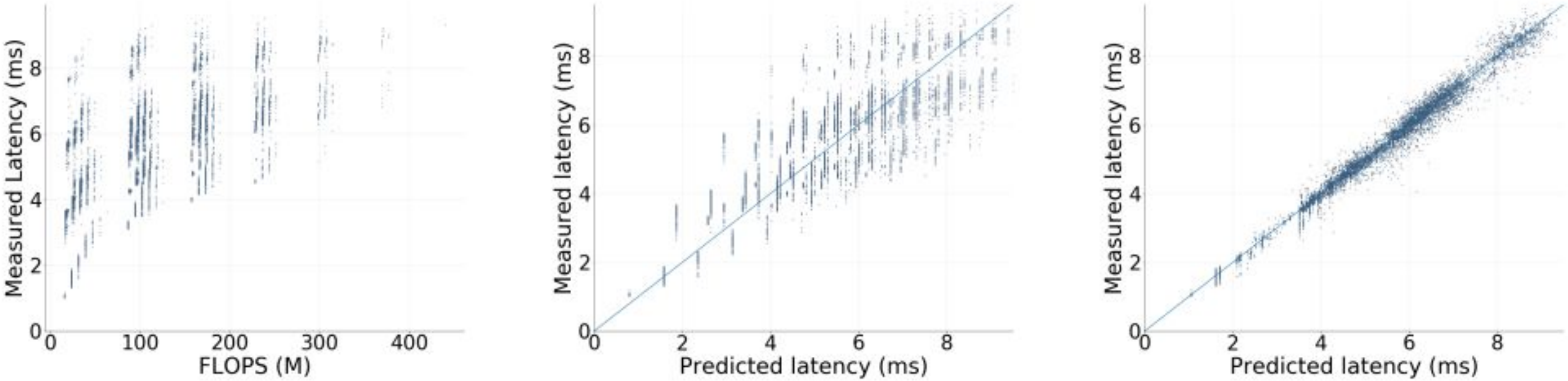


Graph Convolutional Networks (GCN) predictor



Latency prediction with GCN

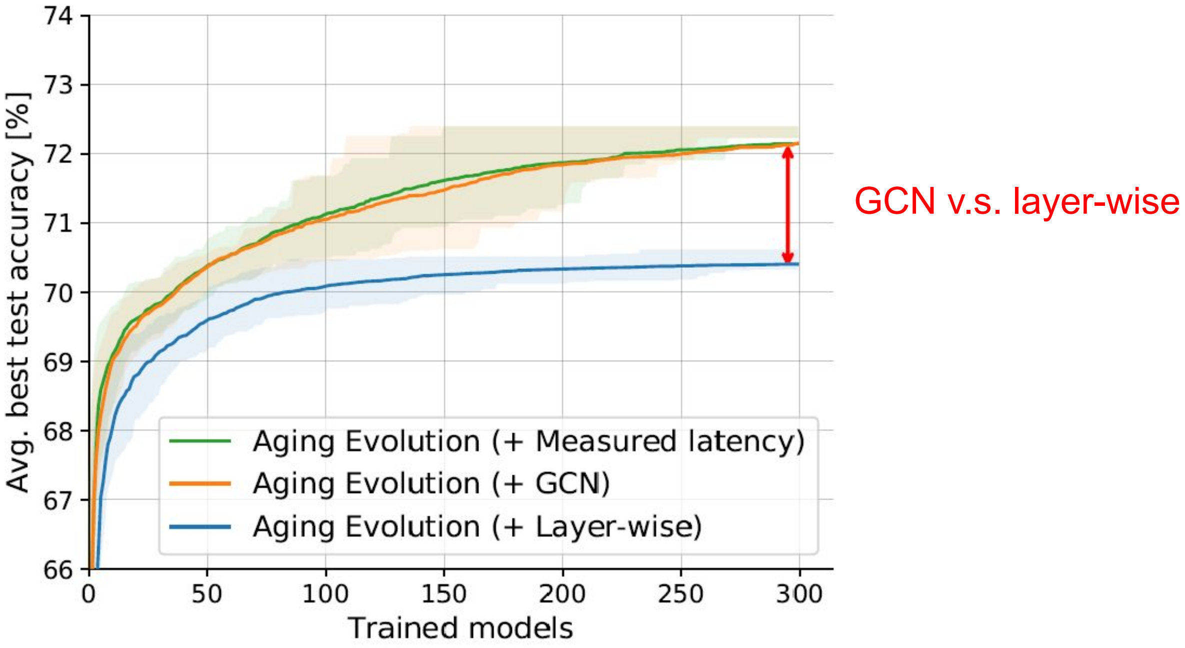
- Significantly outperforms conventional approaches



FLOPs Layer-wise GCN

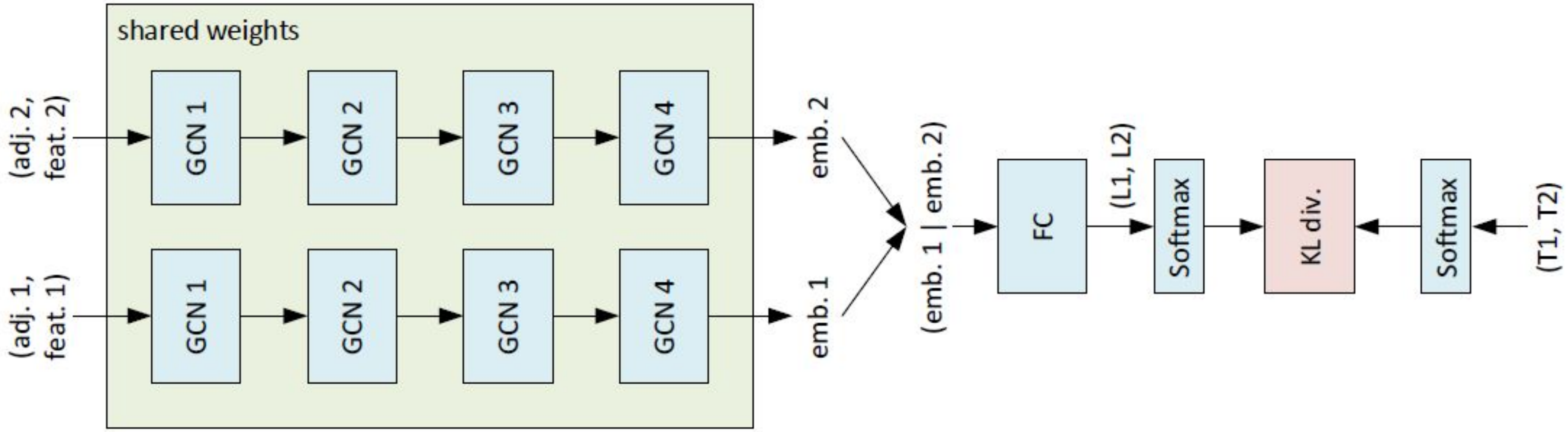
Latency prediction with GCN in NAS

- More accurate models found given latency constraint

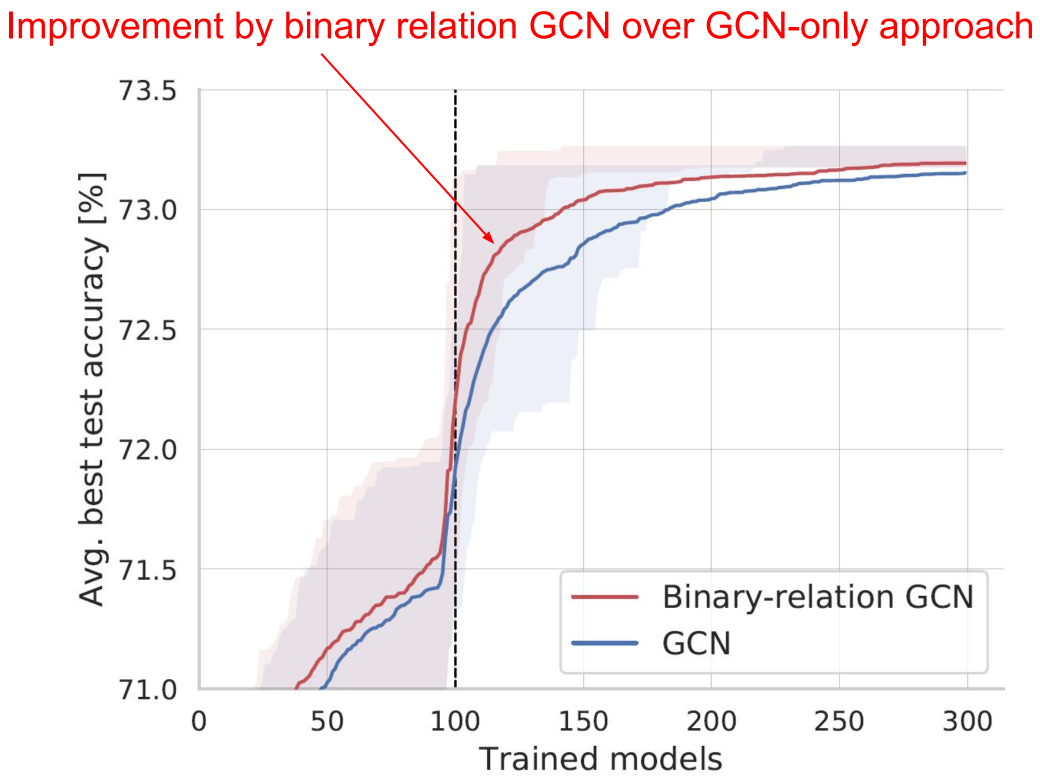


Accuracy prediction with GCN

- ***Binary relation*** improves the accuracy ranking of neural models.
- ***Ranking*** is more important than the absolute values.

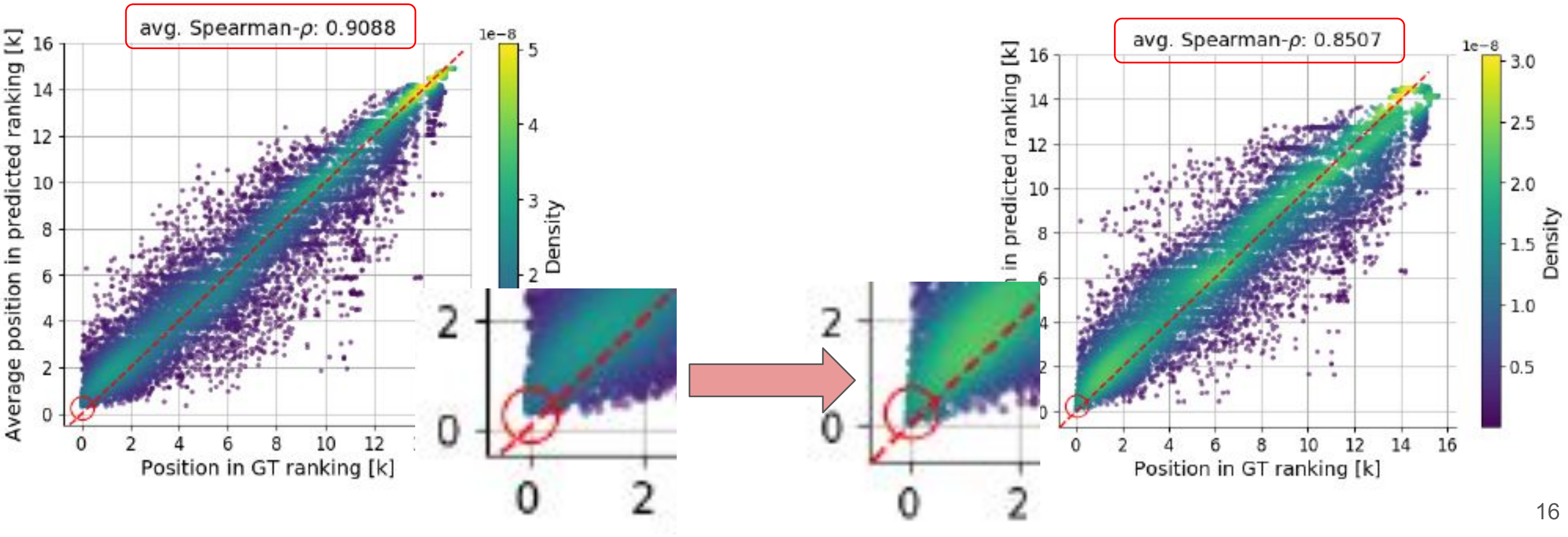


Binary relation GCN vs vanilla GCN

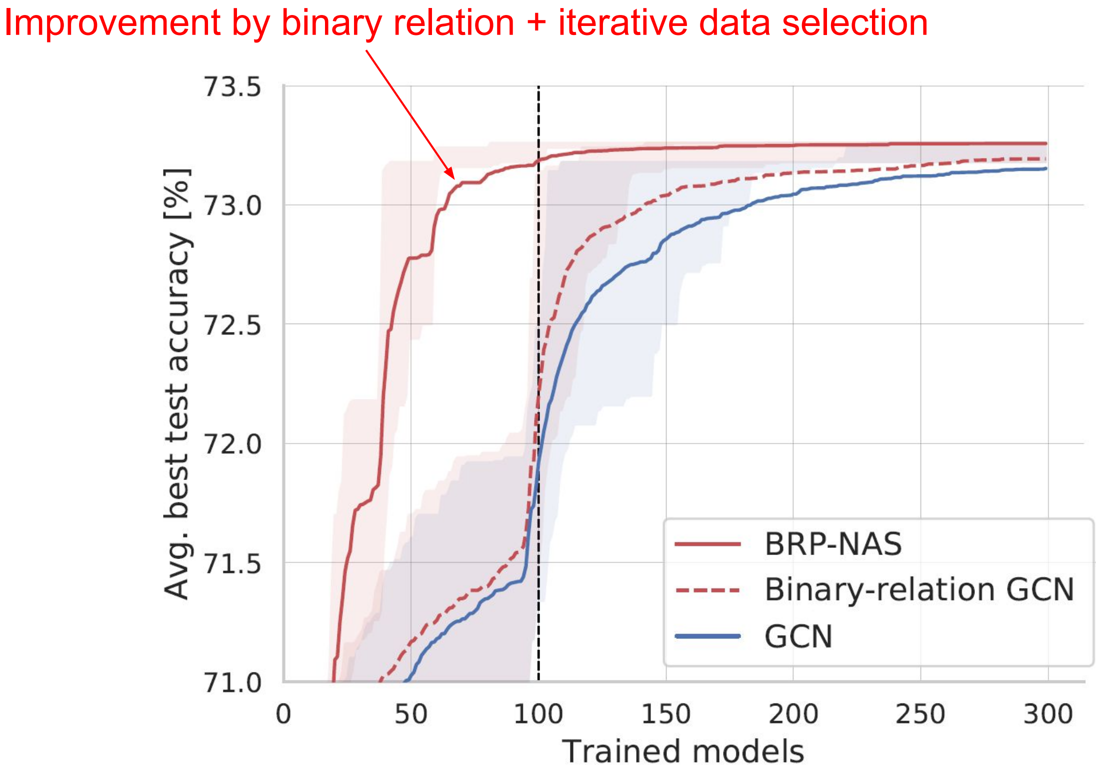


Iterative data selection

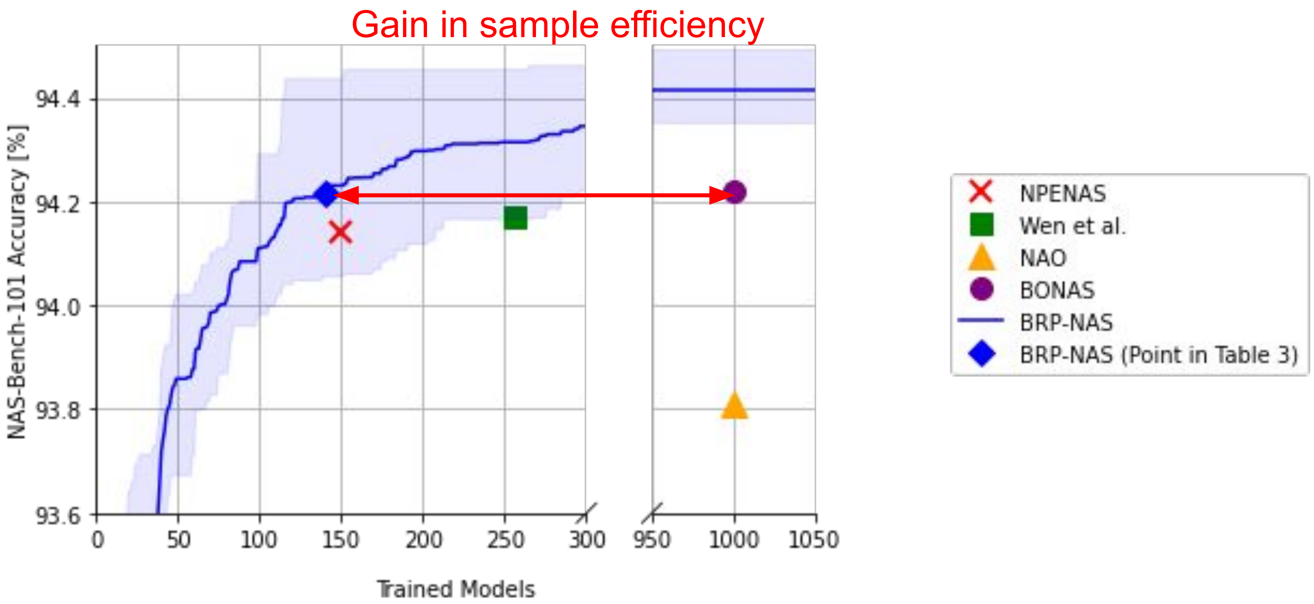
- Train the binary relation predictor iteratively
- Gradually focus on ***higher ranking models*** rather than the global landscape



Iterative data selection + binary relation prediction



Improved sample efficiency



It’s open source!

https://github.com/SamsungLabs/eagle

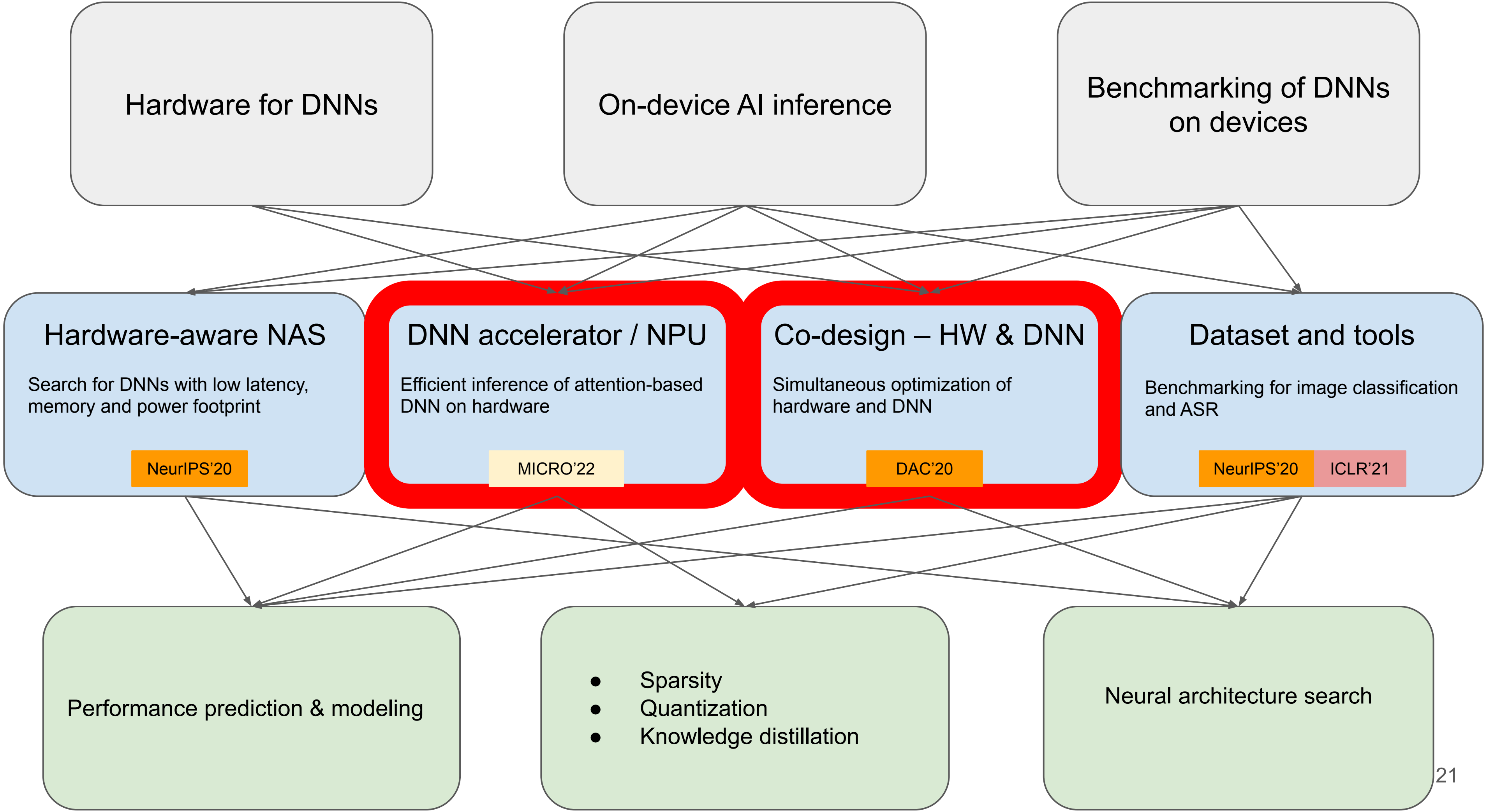
eagle Public

Measuring and predicting on-device metrics (latency, power, etc.) of machine learning models

Python · Apache-2.0 · 8 · 39 · 0 · 0 · Updated on Feb 2

- Dataset
  - Latency of NAS-Bench-201 CNN models
  - Devices: Intel Core i7-7820X, NVIDIA GTX 1080 Ti, NVIDIA Jetson Nano, Google EdgeTPU, Qualcomm Adreno 612 GPU, Qualcomm Hexagon 690 DSP.
- Benchmarking tool
  - Modularised to allow extension to different models and devices

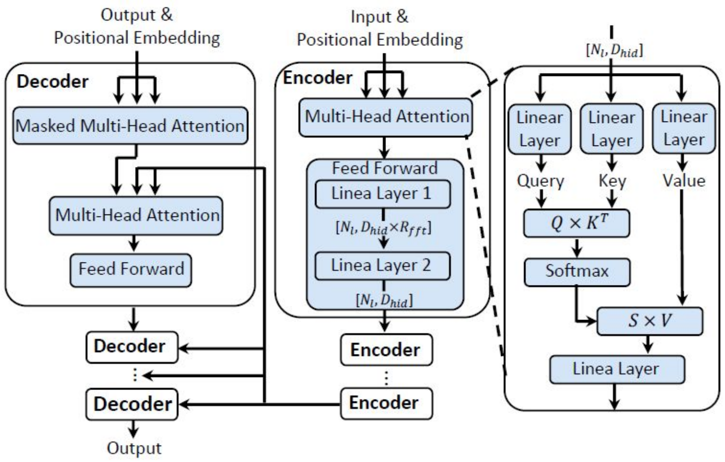
ML accelerator / NPU for Attention-based NNs


Hardware for DNNs
On-device AI inference
Benchmarking of DNNs on devices
Hardware-aware NAS
Search for DNNs with low latency, memory and power footprint
NeurIPS'20
DNN accelerator / NPU
Efficient inference of attention-based DNN on hardware
MICRO'22
Co-design – HW & DNN
Simultaneous optimization of hardware and DNN
DAC'20
Dataset and tools
Benchmarking for image classification and ASR
NeurIPS'20
ICLR'21
Performance prediction & modeling
Sparsity
Quantization
Knowledge distillation
Neural architecture search

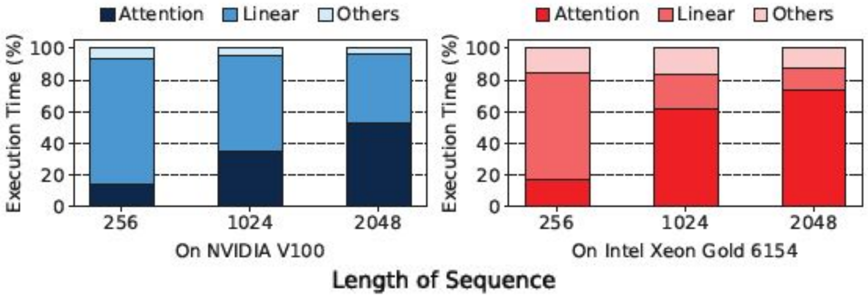
Transformers — opportunities and challenges

Opportunities
● Persuasive in many AI tasks ● Excellent algorithmic performance

Challenges
● Heavy computation and memory resource ● Lacking hardware efficient implementation



Attention and feed forward layers demands excessive computation and memory resources that compromises hardware performance.



Latency is dominated by different components depending on the length of input sequences.

Hardware efficient attention-based NNs

Neural network

- A unified sparsity pattern
- Compress both attention and linear layers

FFT-Attention-Butterfly Network

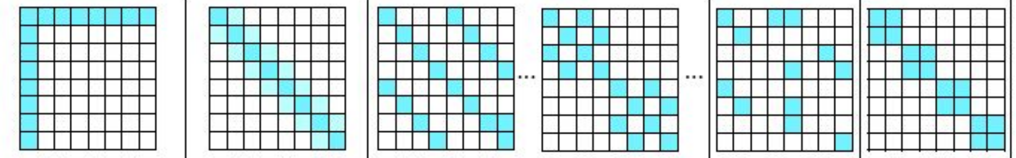
Hardware

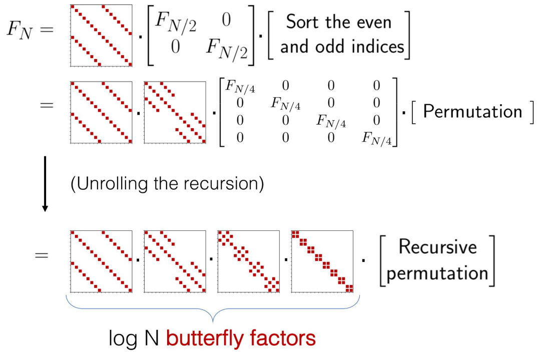
- A unified compute engine
- Reconfigurable at runtime to accelerate different layers

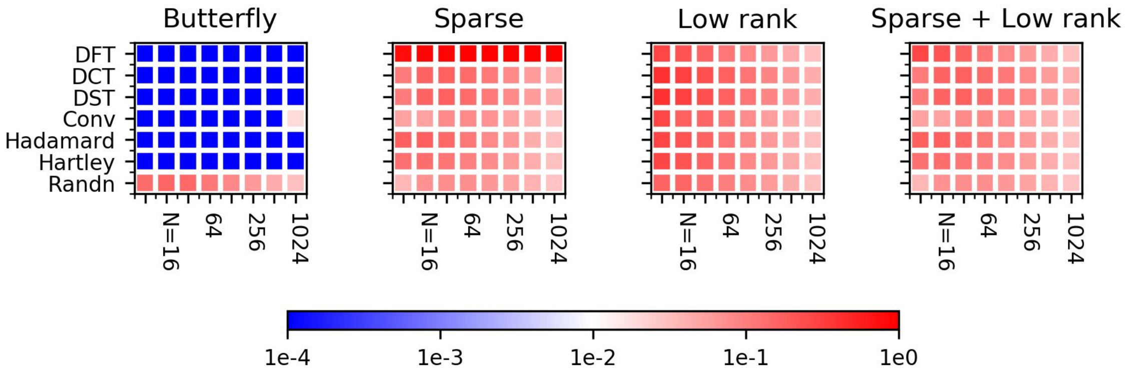
FFT-Attention-Butterfly Engine

Compression — Sparsity patterns

- A universal building block that captures general structured matrices
  - Approximate linear transformations
  - Factor matrices into products of log(N) matrices with a small total number of nonzeros.

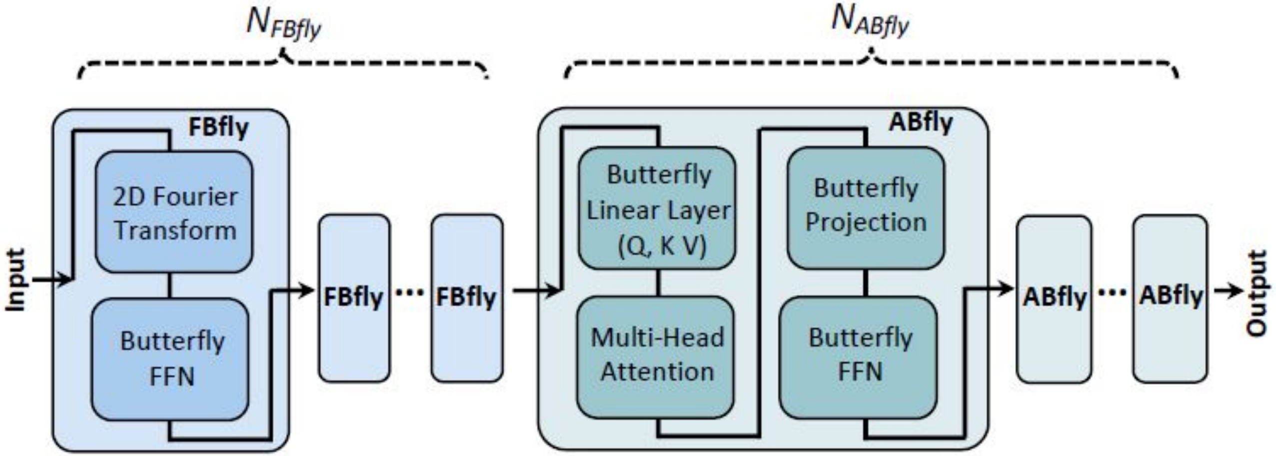
Sparsity	Low Rank	Sliding Window	Butterfly	Random	Block-wise
Patten					
Data Access	Sequential row & column read	Regular stride read	Regular stride read	Random read	Regular stride read
HW Eff.	No	Yes	Yes	No	Yes
Info.	Global	Local	Global & Local	Global&Local	Local

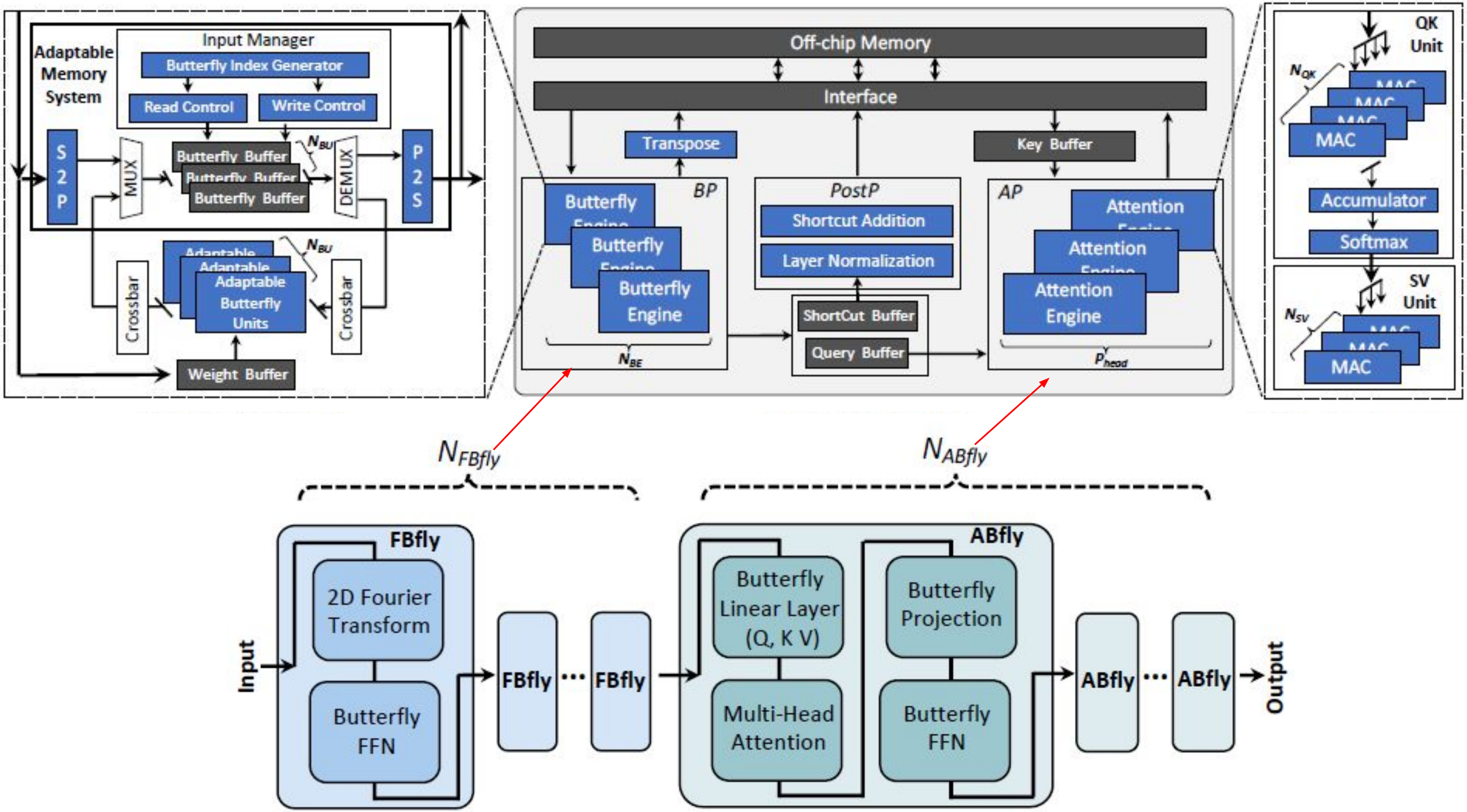




FABNet: FFT-Attention-Butterfly

- Adopts butterfly sparsity pattern to approximate both the attention mechanism and the FFNs
- Replace self-attention modules by 2D Discrete Fourier Transform (DFT) operations





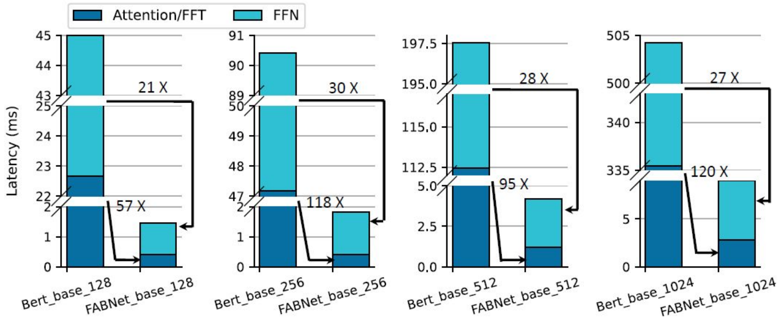
Adaptable Memory System
Input Manager
Butterfly Index Generator
Read Control
Write Control
S 2 P
MUX
Butterfly Buffer
DEMUX
P 2 S
N_{BU}
Adaptable Butterfly Units
Crossbar
Weight Buffer
Off-chip Memory
Interface
Transpose
Key Buffer
BP
Butterfly Engine
N_{BE}
PostP
Shortcut Addition
Layer Normalization
ShortCut Buffer
Query Buffer
AP
Attention Engine
P_{head}
QK Unit
N_{QK}
MAC
Accumulator
Softmax
SV Unit
N_{SV}
N_{FBfly}
N_{ABfly}
FBfly
2D Fourier Transform
Butterfly FFN
ABfly
Butterfly Linear Layer (Q, K V)
Multi-Head Attention
Butterfly Projection
Butterfly FFN
Input
FBfly ··· FBfly
ABfly ··· ABfly
Output

Accuracy

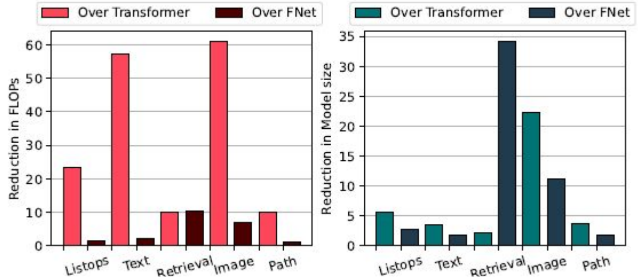
Accuracy of different models on LRA.

	ListOps	Text	Retrieval	Image	Pathfinder	Avg.
Vanilla *Transformer*	0.373	**0.637**	0.783	0.379	**0.709**	**0.576**
Vanilla *FNet*	0.365	0.630	0.779	0.288	0.66	0.544
FABNet	**0.378**	0.630	**0.800**	**0.399**	0.674	**0.576**

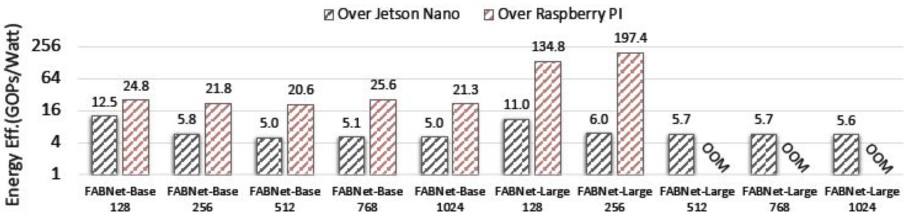
Latency



Model size / FLOPs



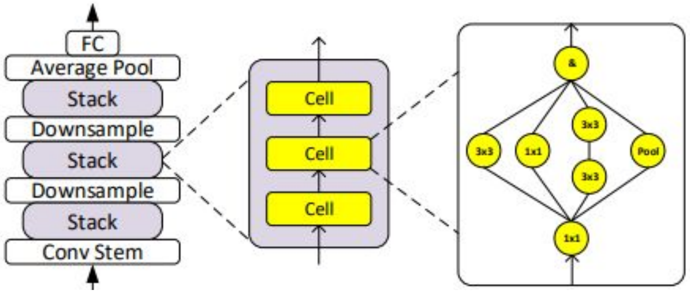
Energy

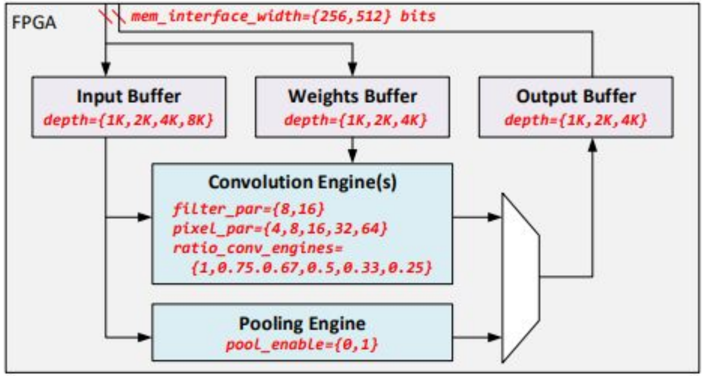


Accelerators	A^3 [17] (HPCA'20)	SpAtten [39] (HPCA'21)	Sanger [25] (MICRO'21)	Energon [44] (TCAD'21)	ELSA [18] (ISCA'21)	DOTA [29] (ASPLOS'22)	FTRANS [24] (ISLPED'20)	Our work
Technology	ASIC (40nm)	ASIC (40nm)	ASIC (55nm)	ASIC (45nm)	ASIC (40nm)	ASIC (22nm)	FPGA (16nm)	FPGA (16nm)
Frequency	1 GHz						170 MHz	200 MHz
# of Multipliers	128						6531	640
Latency (ms)	56.0	48.8	45.2	44.2	34.7	34.1	61.6	**2.4**
Power (W)	1.217	1.060	0.801	2.633	0.976	0.858	25.130	10.727
Energy Eff. (Pred./J)	14.67	19.33	27.62	8.59	29.52	34.18	0.65	**40.48**

Co-design of DNN and hardware

- Automate HW-NN co-design using NAS
  - Include both the DNN and HW parameters
  - Multi-objective optimization: search for best model-HW pair that boosts accuracy and efficiency.

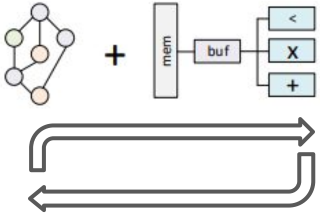




Model

Device

Different model and hardware to meet different application requirements



Accuracy | Latency

Memory | Energy

Effectiveness of Co-design automation

- For image classification, we enumerate close to 4 billion (423k x 8.6k) model-accelerator pairs

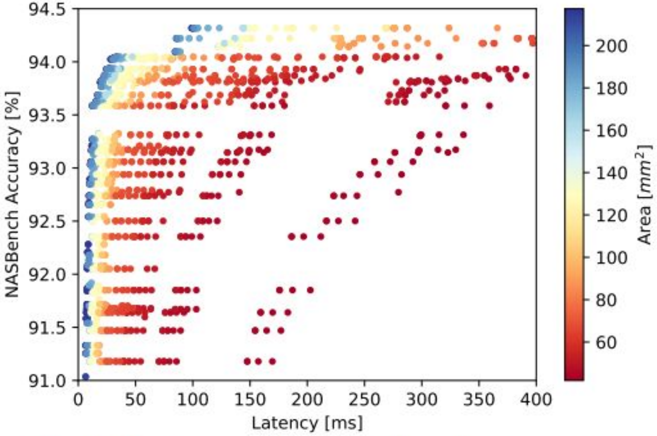


Fig. 4: Pareto-optimal points in the codesign search space.

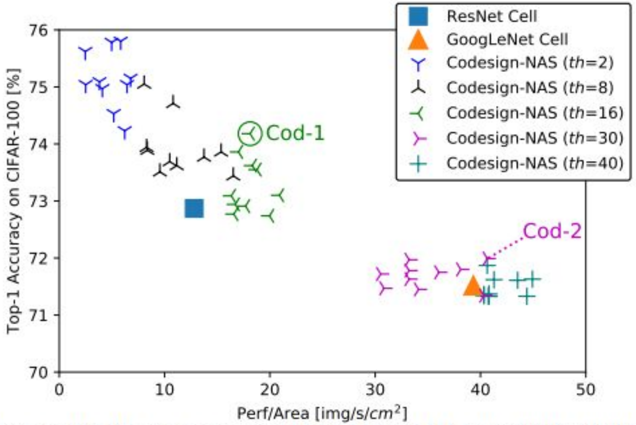
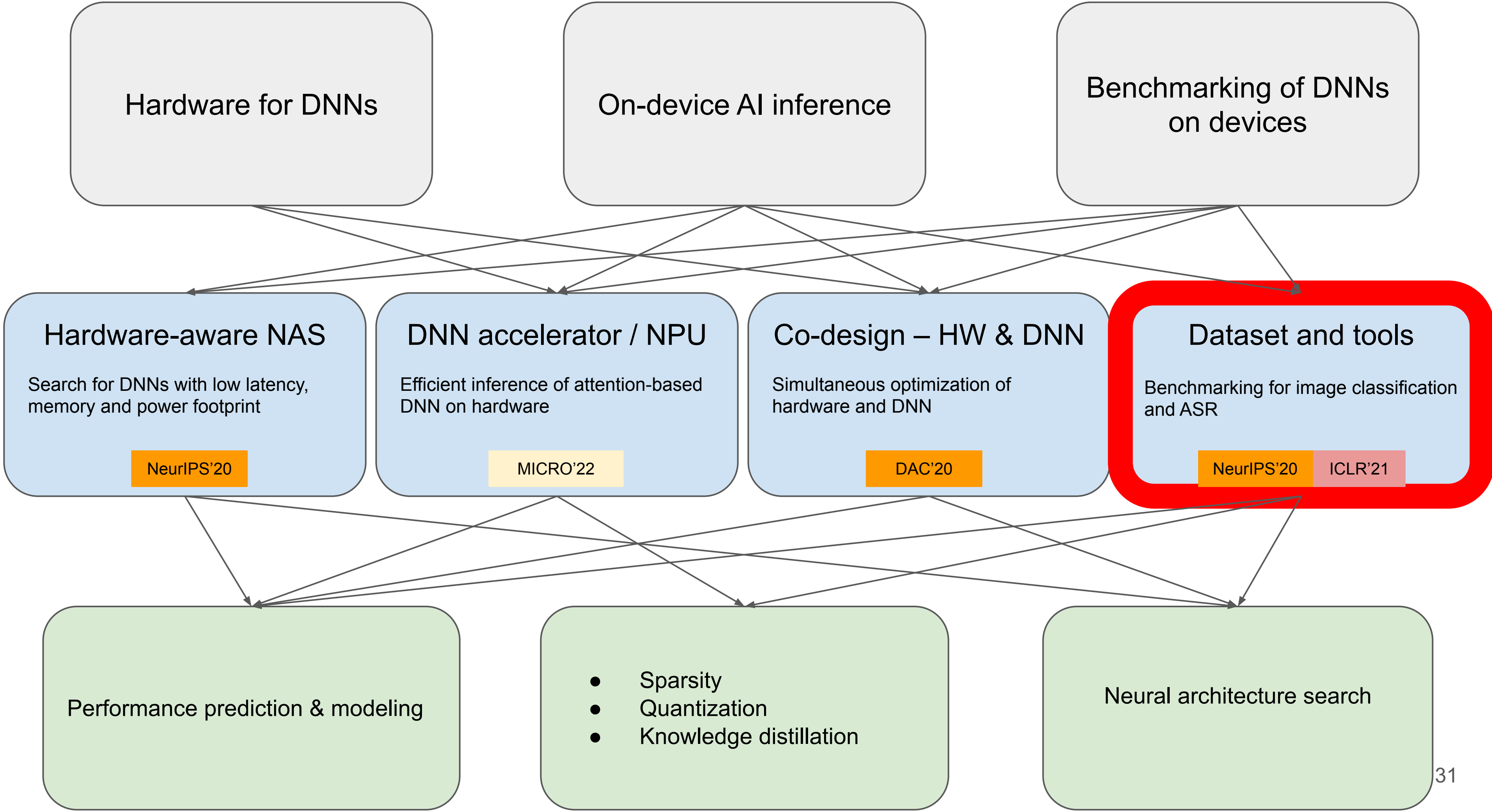


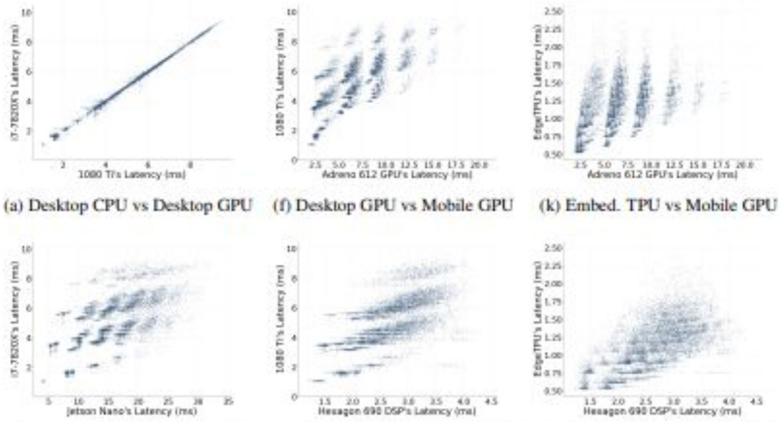
Fig. 7: Top 10 points discovered by Codesign-NAS at each threshold value, compared to ResNet and GoogLeNet cells.

CNN	Accuracy [%]	Perf/Area [$img/s/cm^2$]	Latency [ms]	Area [mm^2]
ResNet Cell	72.9	12.8	42.0	186
Cod-1	**74.2** (+1.3%)	**18.1** (+41%)	**41.8** (-0.5%)	**132** (-29%)
GoogLeNet Cell	71.5	39.3	19.3	**132** (-0.8%)
Cod-2	**72.0** (+0.5%)	**40.6** (+3.3%)	**18.5** (-4.2%)	133

Dataset and Tools


Hardware for DNNs
On-device AI inference
Benchmarking of DNNs on devices
Hardware-aware NAS
Search for DNNs with low latency, memory and power footprint
NeurIPS'20
DNN accelerator / NPU
Efficient inference of attention-based DNN on hardware
MICRO'22
Co-design – HW & DNN
Simultaneous optimization of hardware and DNN
DAC'20
Dataset and tools
Benchmarking for image classification and ASR
NeurIPS'20
ICLR'21
Performance prediction & modeling
Sparsity
Quantization
Knowledge distillation
Neural architecture search

LatBench - Latency dataset of CNN models

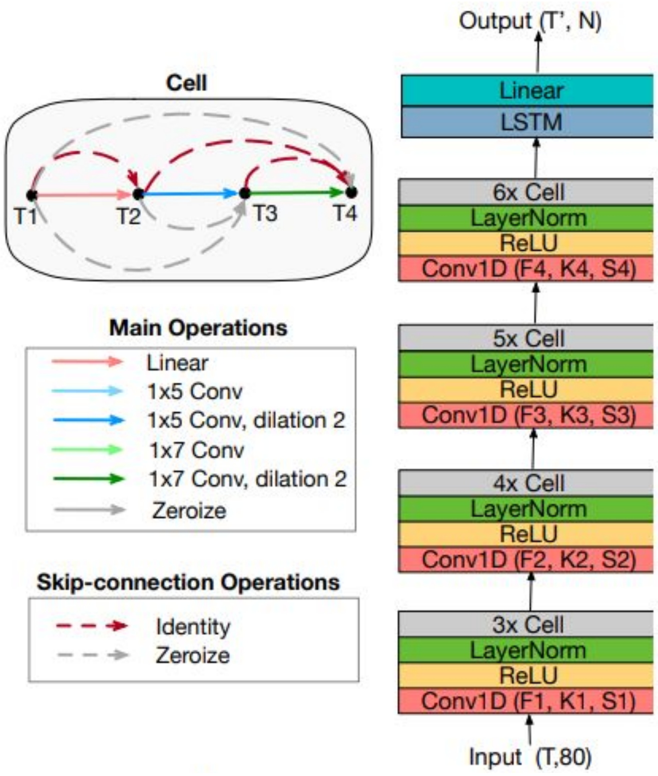


(a) Desktop CPU vs Desktop GPU (f) Desktop GPU vs Mobile GPU (k) Embed. TPU vs Mobile GPU

(b) Desktop CPU vs Embed. GPU (g) Desktop GPU vs Mobile DSP (l) Embed. TPU vs Mobile DSP

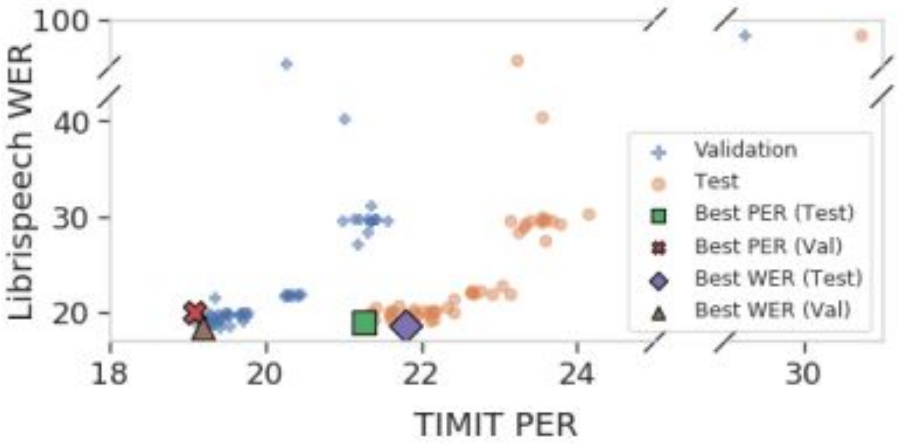
Number of models	15,625
Dataset	CIFAR-100
Metrics	• Validation and Test loss • Number of parameters • FLOPs • Latency
Devices	• Intel Core i7-7820X • NVIDIA GTX 1080 Ti • NVIDIA Jetson Nano • Google EdgeTPU • Qualcomm Adreno 612 GPU • Qualcomm Hexagon 690 DSP

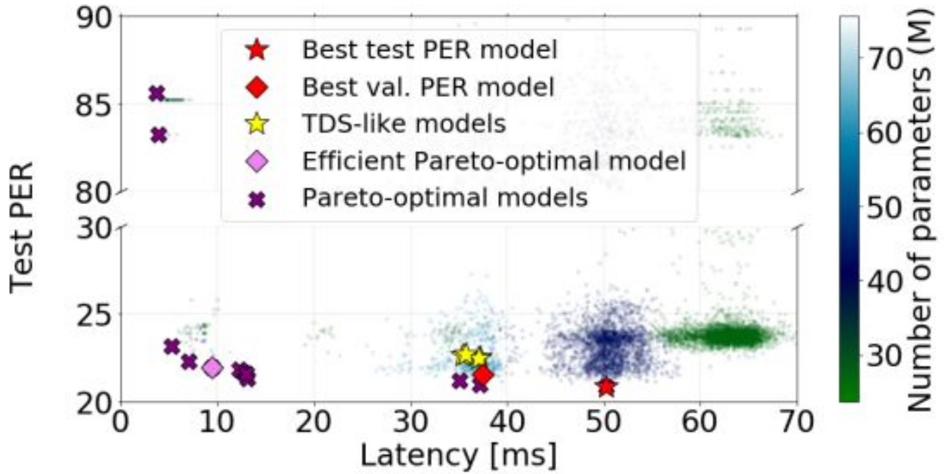
NAS benchmark for ASR models



Number of models	8,242
Target epochs	5, 10 and 40
Dataset	TIMIT
Metrics	• Validation and Test PERs • CTC loss • Number of parameters • FLOPs • Latency (Tesla 1080Ti, Jetson Nano)

NAS benchmark for ASR models





https://github.com/SamsungLabs/nb-asr

Thank you!

- BRP-NAS: Prediction-based NAS using GCNs
  - https://arxiv.org/pdf/2007.08668.pdf
- Best of Both Worlds: AutoML Codesign of a CNN and its Hardware Accelerator
  - https://arxiv.org/pdf/2002.05022.pdf
- NAS-Bench-ASR: Reproducible Neural Architecture Search for Speech Recognition
  - https://openreview.net/pdf?id=CU0APx9LMaL
- Adaptable Butterfly Accelerator for Attention-based NNs via Hardware and Algorithm Co-design