Profiling Neural Blocks and Design Spaces for Mobile Neural Architecture Search

Keith G. Mills[1], Fred X. Han[2], Jialin Zhang[3], Seyed Saeed Changiz Rezaei[2], Fabian Chudak[2], Wei Lu[2], Shuo Lian[3], Shangling Jui[3] and Di Niu[1]

[1]University of Alberta

[2]Huawei Technologies Canada Co., Ltd.

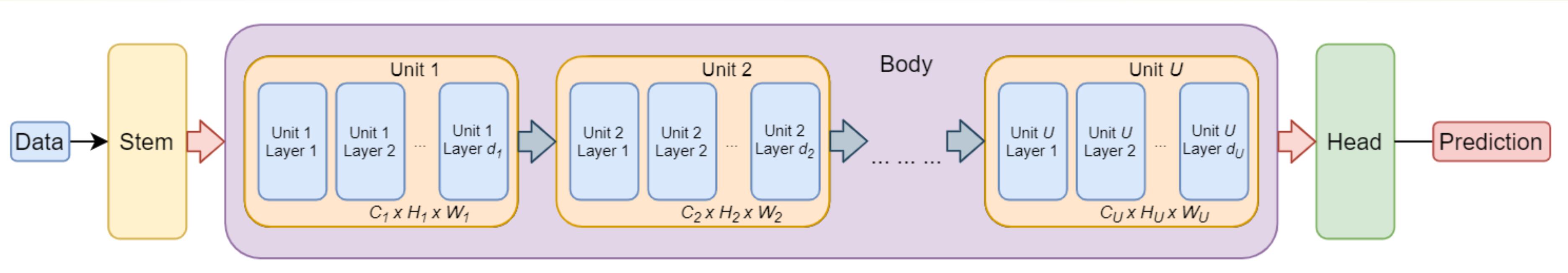
[3]Huawei Kirin Solution, Shanghai, China

Open-Source Repo: https://github.com/Ascend-Research/BlockProfile

Motivation

- Neural Architecture Search:
  - Three core components [Elsken et al., 2019]:
    - Design Space/Search Space – Set of all possible candidate architectures, e.g., MobileNetV2 [Sandler et al., 2019] or V3 [Howard et al., 2019].
    - Search Algorithm – Traverses the Search Space. *Many* forms, e.g., Evolutionary Algorithms, Reinforcement Learning, Gradient Descent, Bayesian Optimization, etc.
    - Performance Estimation Strategy – How an architecture is evaluated, e.g., train all models from scratch or a weight-sharing supernet [Cai et al., 2019, Cai et al., 2020].
  - Vast amount of literature on developing better Search Algorithms and more accurate Performance Estimation Strategies.
  - Less work devoted to Design Spaces, but it is nevertheless important:
    - Resource usage concerns and the need to be *hardware-friendly* [Cai et al., 2020]
    - Inference latency is *not* consistent across varying hardware.
  - **Our objective:** Profile well-known Design Spaces for accuracy/latency on different hardware.

What Is A Design Space?



- Components of the neural network that can be adjusted, i.e., searched.
  - Most commonly, this is the network body, but not the stem or head.
- Abstract body structure into 3 levels of increasing granularity:
  - Units, u, which perform operations on unique tensor sizes.
  - Layers, l, a varying number within a given unit.
  - Operation Blocks, b, which perform computation.
  - Notation: Block b at layer l of unit u can be denoted with the tuple (u, l, b).

What Real Design Spaces Look Like

- Once-for-All (MobileNetV3) [Cai et al., 2020], denoted OFA
  - 5 units, 2-4 layers/unit
  - MBConv Blocks with adjustable expansion/kernel size
  - Variable input resolution {224, 208, 192,…}
- ProxylessNAS (MobileNetV2) [Cai et al., 2019], denoted PN
  - 6 units, 2-4 layers/unit in first 5. Final unit contains only 1 layer.
  - MBConv Blocks with adjustable expansion/kernel size
- ResNet50 [He et al., 2016]
  - 4 units.
    - Units 1, 2 and 4 have 2-4 layers
    - Unit 3 has 4-6 layers
  - Unit-wide channel expansion ratios
  - Layer expansion ratio for blocks.
  - Operations consist of simple 1x1 and 3x3 convolutions, not searchable.

Table 1: Candidate blocks for MobileNets (OFA and ProxylessNAS; left) and ResNet50 (right). Blk. Code is a proxy name we use for figures in Section 4.2 to simplify notations.

MobileNets	Exp. Ratio	Kernel Size	Blk. Code	ResNet50	Unit Ratio	Layer Ratio	Blk. Code
MBConv3–3	3	3×3	B1	65–0.20	0.65	0.20	C65–B20
MBConv3–5	3	5×5	B2	65–0.25	0.65	0.25	C65–B25
MBConv3–7	3	7×7	B3	65–0.35	0.65	0.35	C65–B35
MBConv4–3	4	3×3	B4	80–0.20	0.8	0.20	C80–B20
MBConv4–5	4	5×5	B5	80–0.25	0.8	0.25	C80–B25
MBConv4–7	4	7×7	B6	80–0.35	0.8	0.35	C80–B35
MBConv6–3	6	3×3	B7	100–0.20	1.0	0.20	C100–B20
MBConv6–5	6	5×5	B8	100–0.25	1.0	0.25	C100–B25
MBConv6–7	6	7×7	B9	00–0.35	1.0	0.35	C100–B35

How Do We Profile a Design Space?

Assume we're talking about one search space at a time:

- Let A denote a uniformly sampled architecture in terms of the number of layers in each unit, block assignment per layer.
- Now let $A_{(u,\,l,\,b)}$ denote that block b has been assigned to layer l of unit u; that u has at least l layers.
- Evaluate the performance of $A_{(u,\,l,\,b)}$ on a given metric M (e.g., accuracy, latency).
- Sample *many* random architectures (e.g., 100, 100k, 1M), assign (u, l, b) to each and measure. Compute M_b, the expected value of b on metric M on the entire network.

$$M_b = \frac{1}{\sum_{u=1}^{U} d_u} \sum_{u=1}^{U} \sum_{l=1}^{d_u} \mathbb{E}[M(A_{u,l,b})]. \tag{1}$$

Simple, and computationally expensive in absolute terms.

Compared to exhaustive evaluation ($\sim 10^{19}$) for OFA, very cheap.

Block-Wise Performance – OFA MBv3

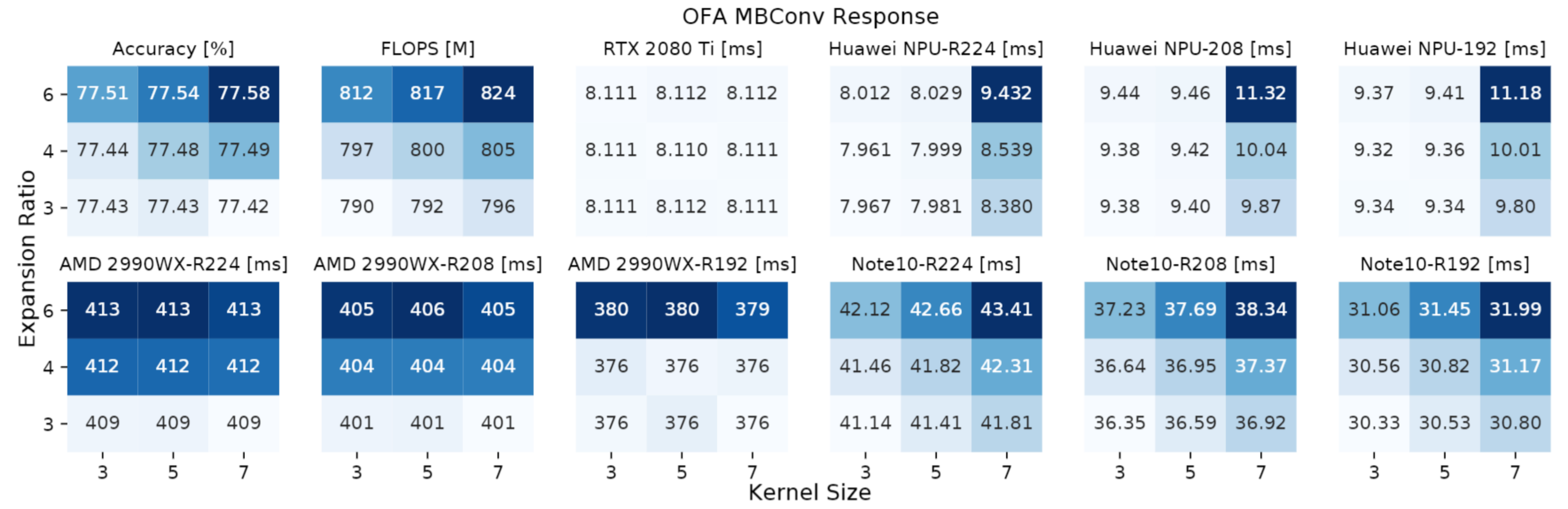


Figure 2: Block-wise average response M_b for OFA-MobileNetV3 blocks in terms of accuracy, FLOPS, and latency on 4 hardware devices. Each entry corresponds to a *MBConv* block identified by an expansion ratio and a kernel size. '-R' flags indicate use of a specific input resolution, assuming 224 by default.

- Accuracy is highly correlated to block size (FLOPS).
- Device optimizations can cause unintuitive trends:
  - Huawei NPU: Kernel size 7 is unfriendly; latency rises as resolution is decreased.
  - Nvidia GPU latency invariant to block size.
  - AMD CPU latency depends on channel expansion ratio.

ProxylessNAS and ResNet50

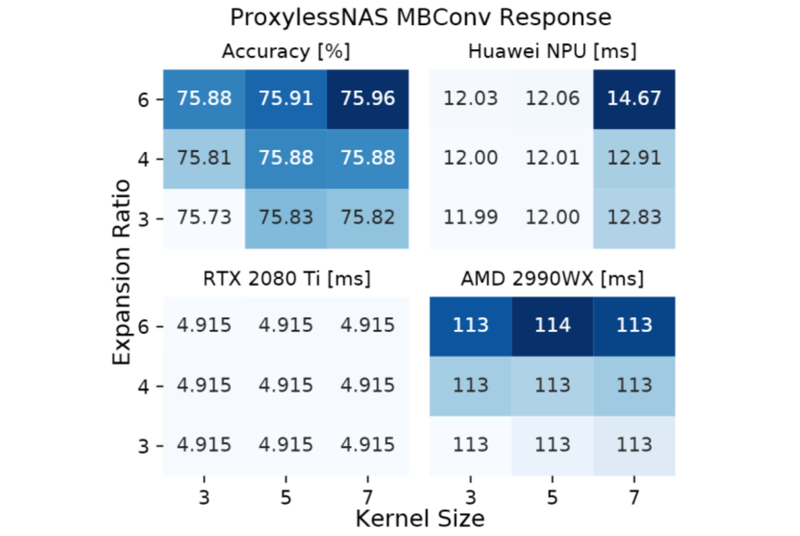


Figure 3: Block-wise average response M_b for blocks in ProxylessNAS on 3 different hardware devices.

- Accuracy: Correlated to block size.
- NPU: Kernel size 7 still unfriendly.
- GPU: Mostly constant
- CPU: Channel dependent

- FLOPS/CPU latency: Depends on units and layers.
- GPU latency is mostly constant with noticeable variation.

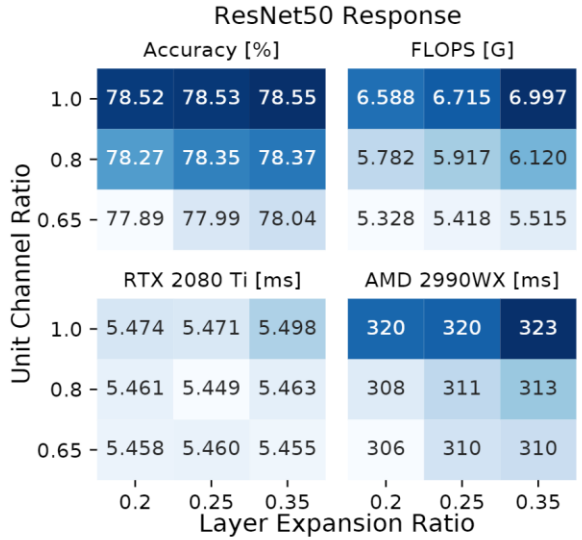


Figure 4: Block-wise average response M_b for blocks in ResNet50 on the GPU and CPU.

Block-Level Performance Is Not Enough

- Recall that different units have different tensor dimensions – height, width, and number of channels.
- The number of layers in a unit is a variable.
- These factors impact block responses differently.
- We characterize the **relative performance** of placing block b in layer l of unit u by calculating $M_{u,l,b}$:

$$M_{u,l,b} = \mathbb{E}(M(A_{u,l,b})) - \mathbb{E}(M(A)), \tag{2}$$

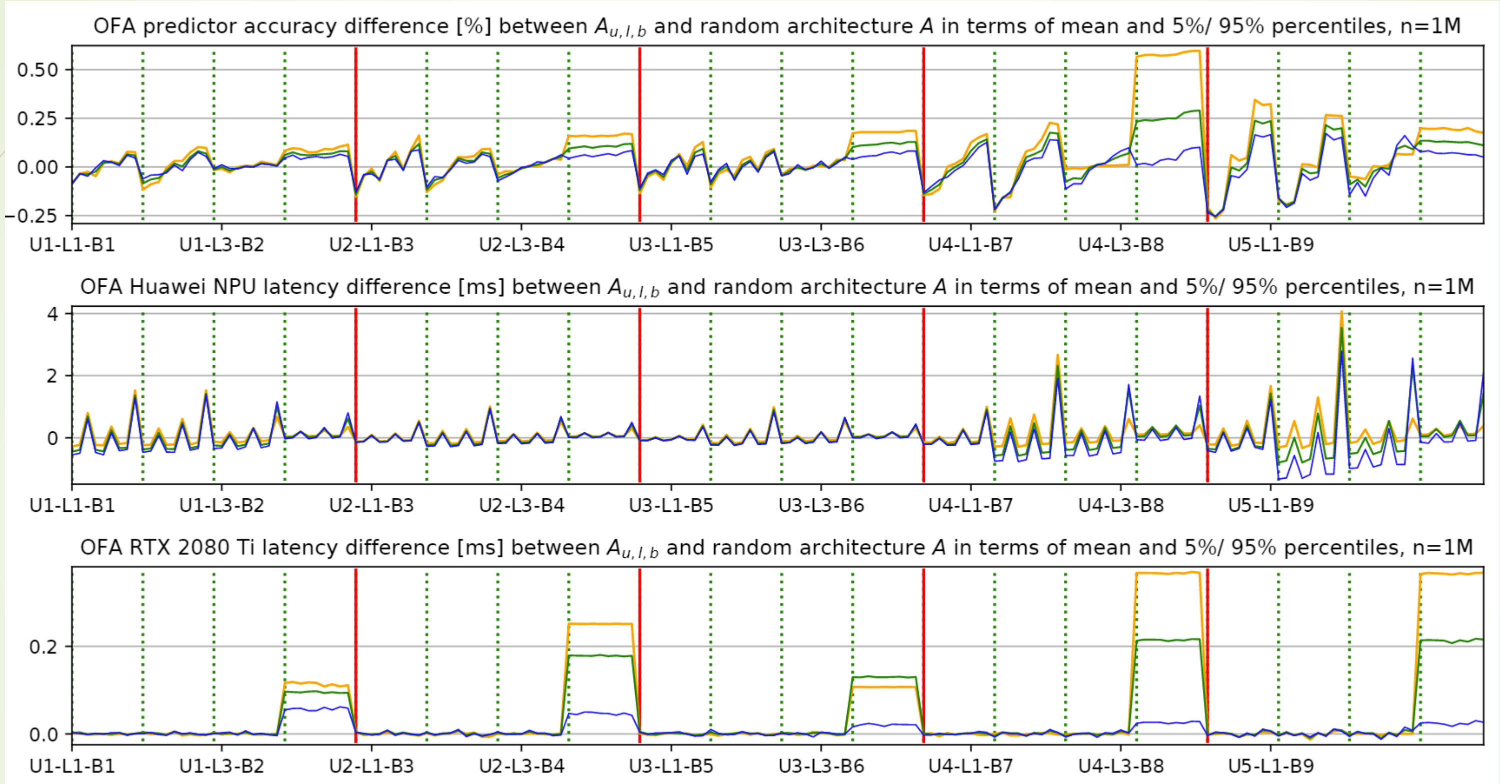
- Can also calculate **relative τ-percentiles**, e.g., *5%* or *95%*:

$$M_{\tau,(u,l,b)} = \mathbb{Q}_\tau(M(A_{u,l,b})) - \mathbb{Q}_\tau(M(A)), \tag{3}$$

UNIVERSITY OF ALBERTA
QUAECUMQUE VERA

HUAWEI

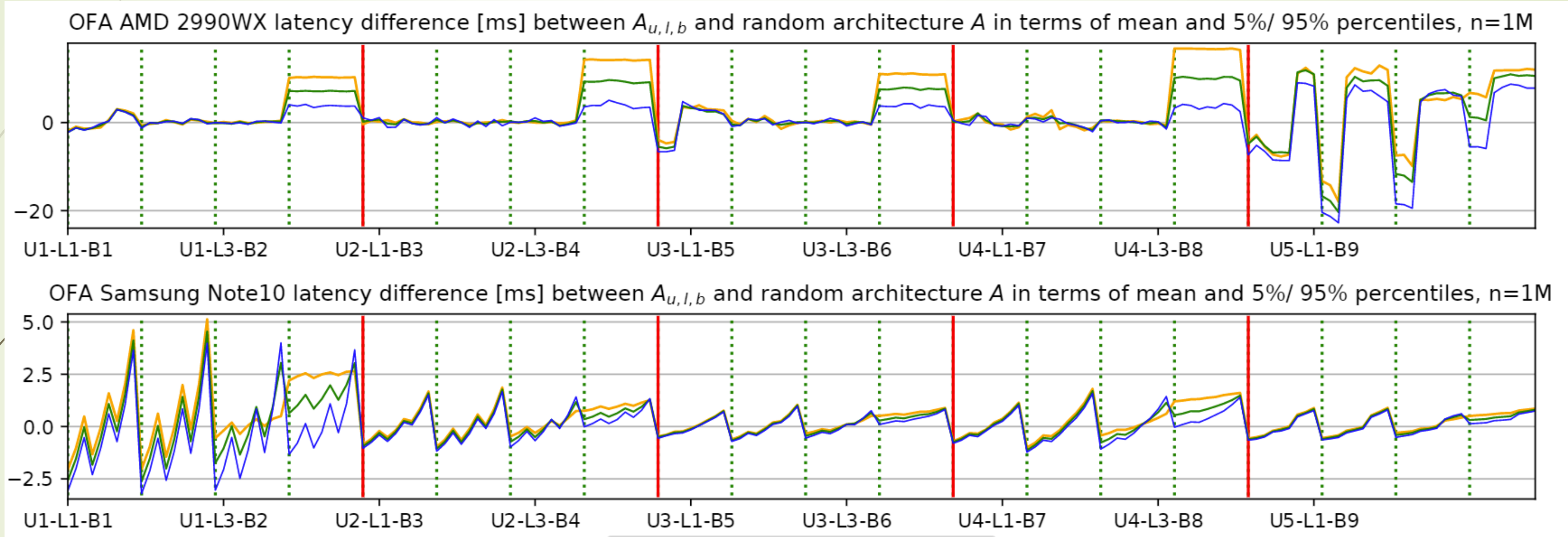
Layer Dependent Performance – Sensitivity on OFA



- Variation due to block choice depends on network depth.
- Accuracy and NPU latency are most sensitive in the final units.
- GPU latency is not sensitive to block choice, but whether a unit has 4 layers.

HUAWEI

Layer Dependent Performance Continued



- CPU latency is sensitive to block choice in the final unit. Otherwise, it depends on whether the unit has 4 layers.
- Note10 latency is most sensitive in the first unit.

Application to NAS – Simple Pruning and Search

- Use insights to reduce the size of the search space and improve accuracy/latency. E.g.,
  - **NPU:** Reduce latency by removing all kernel size 7. Improve accuracy by focusing on final 2 units.
  - **OFA-GPU:** Reduce latency by constraining units 2, 4 and 5 to have at most 3 layers. Increase accuracy by removing low accuracy blocks.
- Benchmark original and pruned spaces on a simple random mutation algorithm.
  - Start from initial pool of random architectures.
  - Apply random perturbations to architectures, generating the next generation.
  - Evaluate, keep the best.
  - Repeat a set number of times.

Pareto Frontier Search

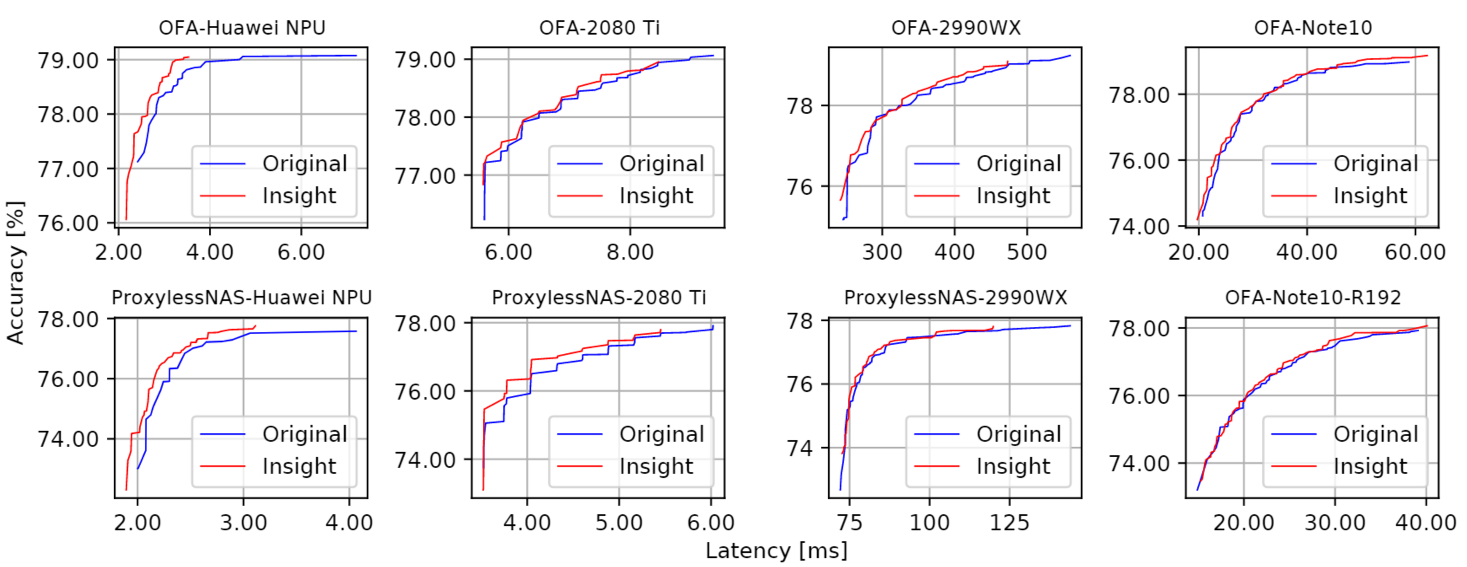


Figure 8: Pareto frontiers contrasting the original search spaces (blue) with our insight-based search spaces (red).

- With pruned search spaces we can find better Pareto frontiers than in the originals.
- Exploit differences in the accuracy and latency distributions.
- Good example: NPU due to kernel size 7 being unfriendly.
- Not observed on devices where latency is highly-correlated to accuracy, like the Samsung Note10.

Maximum Accuracy Search

Table 2: Maximum top-1 ImageNet accuracy search results on different design spaces, compared to existing works. We show averages over 5 random seeds for our experiments.

Model	Accuracy	MACs
MobileNetV2 [21]	72.0	300M
MobileNetV3-Large [10]	75.2	219M
OFA [2]	76.0	230M
OFA_{Large}	79.0	595M
OFA-insight	**79.2** ± 0.04	342M
OFA-base	78.9 ± 0.07	292M
ProxylessNAS-insight	**77.9** ± 0.04	417M
ProxylessNAS-base	77.6 ± 0.08	359M
ResNet50-insight	**80.0** ± 0.03	2.81B
ResNet50-base	79.9 ± 0.09	2.64B

- Remove low accuracy blocks.
- Compete with state-of-the-art.
- Our insights consistently achieve higher results than the original spaces.
- We also outperform the original OFA_{Large} in terms of ImageNet accuracy.

Conclusion

- Our method for profiling mobile blocks
  - allows us to measure accuracy and inference latency and hardware friendliness on the Huawei Kirin 9000 NPU, Nvidia RTX 2080 Ti GPU, AMD Threadripper CPU and Samsung Note10 on the Once-for-All, ProxylessNAS and ResNet50 Design Spaces.
  - can illustrate blockwise performance, which is different for each device.
  - quantify sensitivity to block choice, layers and depth.
  - provides useful information as our gathered insights allow us to prune search spaces, finding better Pareto frontiers and maximum accuracy results that compete with the state-of-the-art.

References

- Elsken, Metzen and Hutter "Neural Architecture Search: A Survey", JMLR.
- Sandler et al. "MobileNetV2: Inverted Residuals and Linear Bottlenecks", CVPR 2018.
- Howard et al. "Searching for MobileNetV3", ICCV 2019.
- Cai, Zhu and Han "ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware", ICLR 2019.
- Cai et al. "Once-for-All: Train One Network and Specialize it for Efficient Deployment", ICLR 2020.
- He et al. "Deep Residual Learning for Image Recognition", CVPR 2016.