IEEE Signal Processing Society

ICASSP2020 Barcelona

MoGA: Searching Beyond MobileNetV3

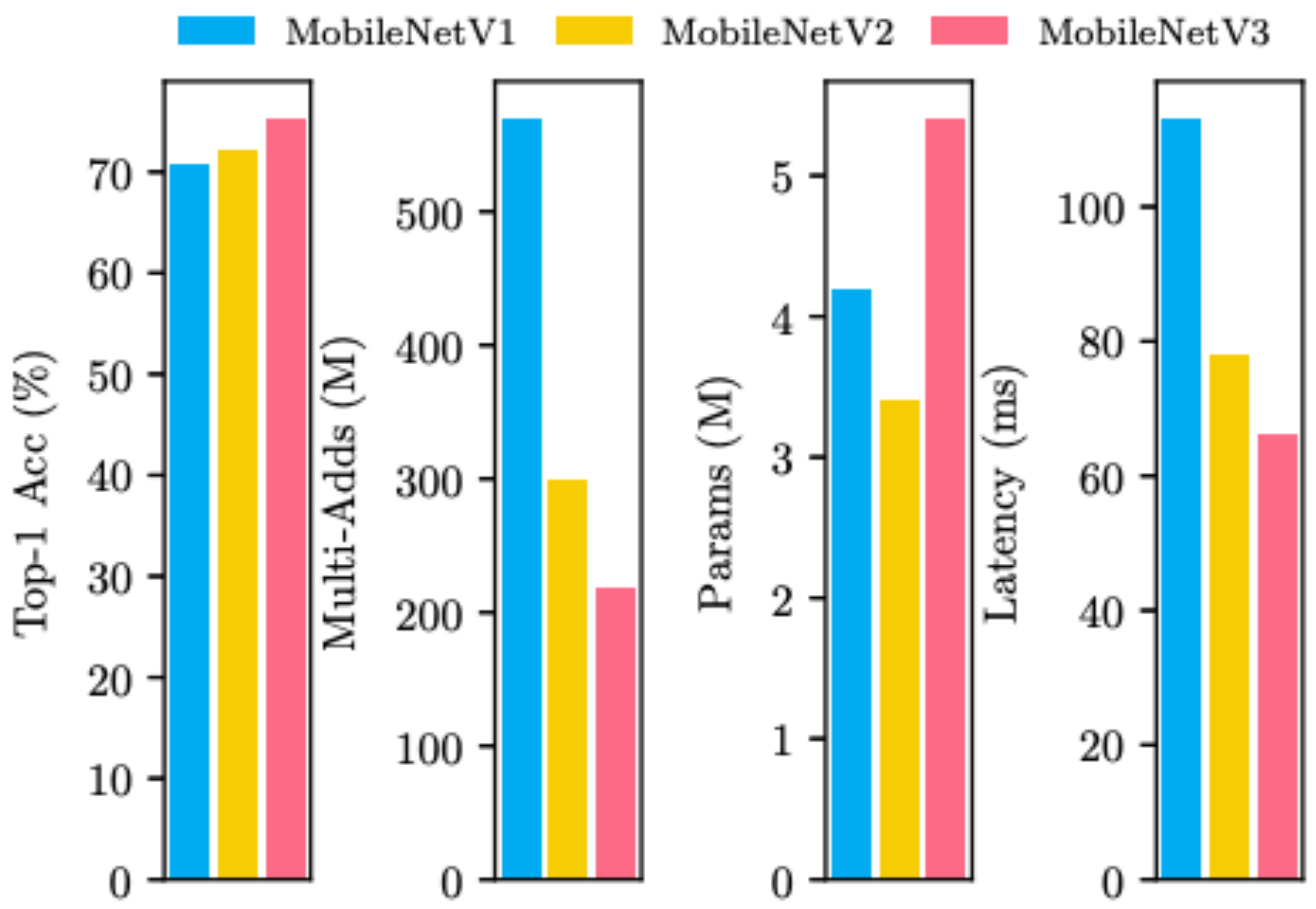
Authors: Xiangxiang Chu, Bo Zhang, Ruijun Xu
Speaker: Bo Zhang
AutoML Team, Xiaomi AI Lab

Trend of Mobile Network Design

- Evolution of MobileNet (Howard et al 2017, Sandler et al. 2018, Howard et al 2019) Series
- Neural Architecture Search (RL/EA, weight-sharing like one-shot, or differentiable)
- Latency-awareness is considered for mobile CPU (Tan et al. 2018)
    - Reward: ACC $\times$(LAT/TAR)^{w}

How MobileNets have changed

- Increased accuracy
- Less Multi-Adds
- **More Parameters**
- Lower latency



Background on Neural Architecture Search

- Search Space Design (Zoph et al. 2017, Tan et al. 2019)
- Searching Algorithms (Reinforcement learning, Evolutionary algorithms, Gradient-based)
- Model Evaluation (Incomplete training, weight-sharing via supernet)

Why Mobile GPU-Awareness (MoGA)?

- Latency is a key factor in mobile applications (e.g. portrait segmentation real-time preview)
- Neural models are deployed usually on mobile GPUs for faster speed, rather than CPU
- CPU is not a good proxy for GPU (low correlation)

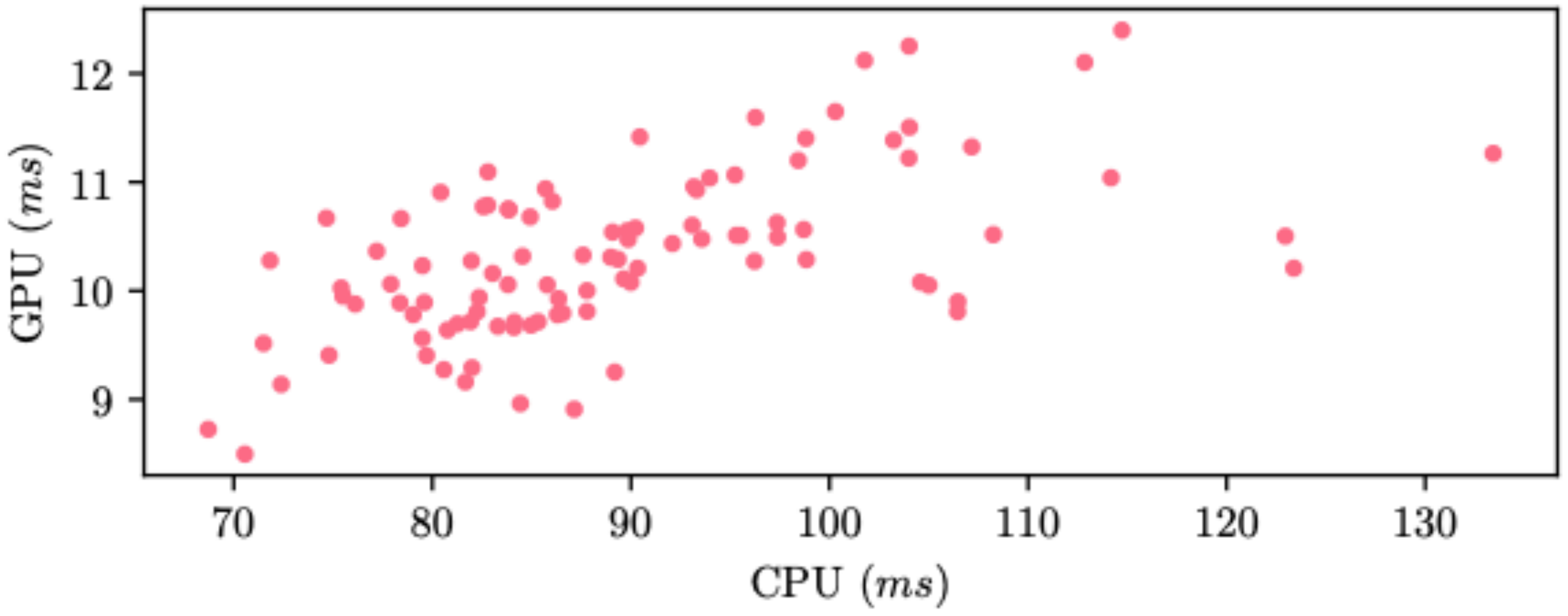


Figure 2: Latency relationship on mobile CPUs vs. on mobile GPUs.

Problem Formulation

- Goal: lower latency, more parameters (to solve underfitting), higher accuracy (MOP)

$$\begin{aligned} &\text{minimize } \{-Acc(m), Lat(m), -Params(m)\}, \forall m \in \Omega \\ &s.t. \quad w_{acc} + w_{lat} + w_{params} = 1, \forall w >= 0 \end{aligned} \tag{3}$$

- Weighted crowding distance in NSGA-II (Deb et al. 2002), care more about *acc*, *lat*, less for *params*

$$D(m_j) = \sum_{i=1}^{n} w_i * \frac{O^i_{neighbor+} - O^i_{neighbor-}}{O^i_{max} - O^i_{min}}. \tag{4}$$

Our Search Space Design

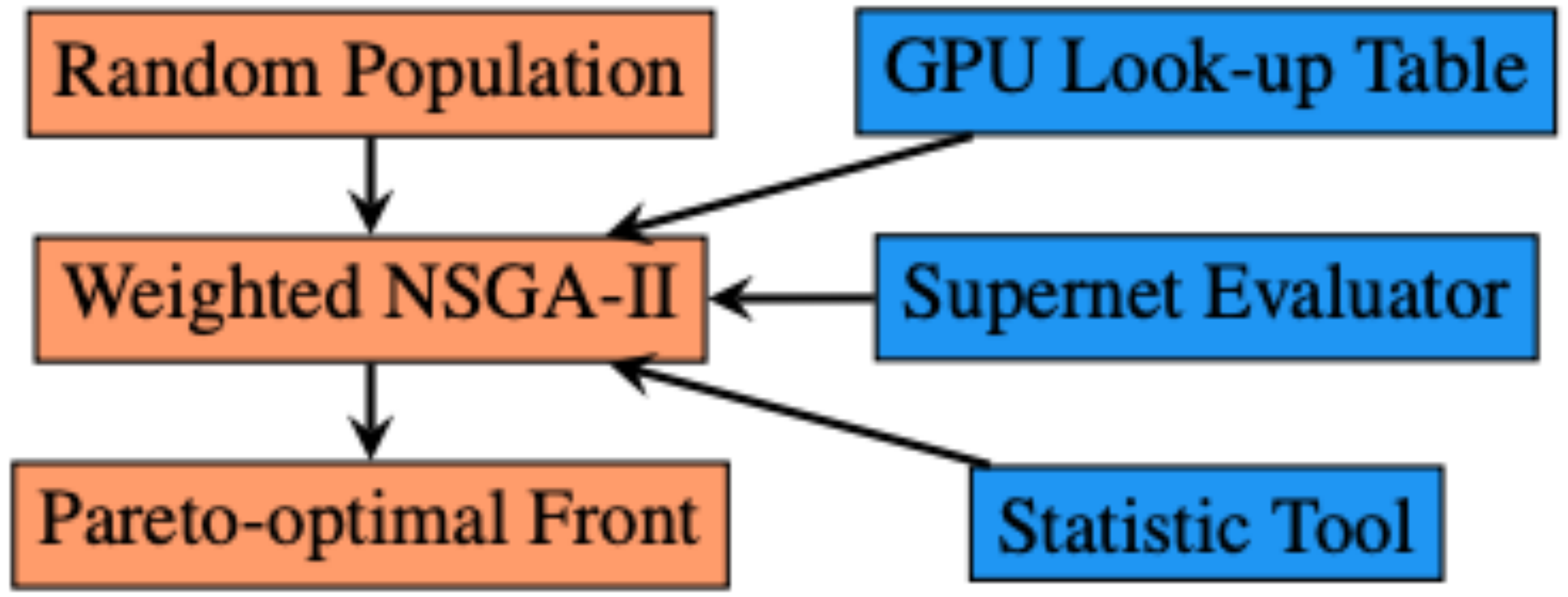
- Built on top of MobileNetV3-Large (Howard el al. 2019)
- 12 choices at block-level
- Size 12^14

Index	Expansion	Kernel Size	SE
0	3	3	-
1	3	3	✓
2	3	5	-
3	3	5	✓
4	3	7	-
5	3	7	✓
6	6	3	-
7	6	3	✓
8	6	5	-
9	6	5	✓
10	6	7	-
11	6	7	✓

Table 1: Each layer in our search space has 12 choices. SE: Squeeze-and-Excitation.

The NAS Workflow

- Train Supernet as in FairNAS (Chu et al. 2019)
- Get each submodel's latency with a look-up table
- Search with weighted NSGA-II until Pareto-optimality



Quick Latency Measurement

- Build a **Latency Lookup Table** based on the cost of each block
- Tool: Mobile AI Compute Engine (MACE) on Mi Mix 3.
- Measurement Accuracy: 0.0571*ms* RMSE

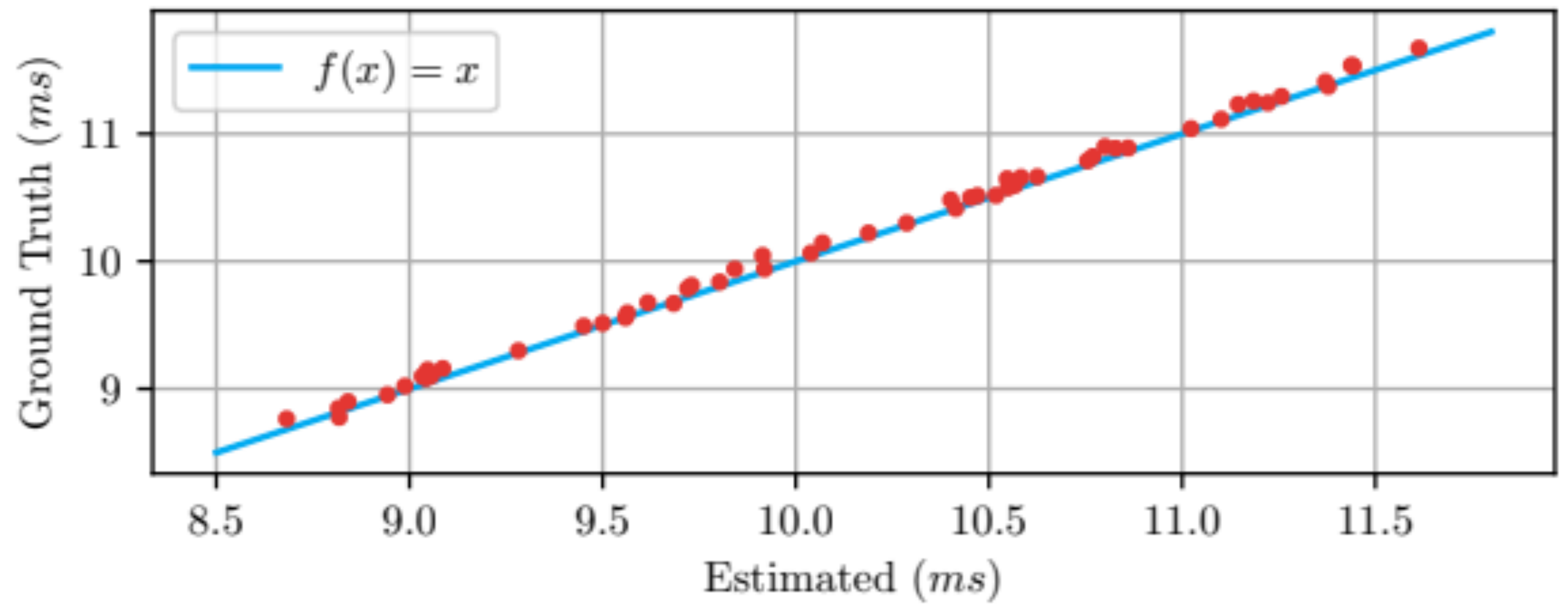


Fig. 3. Mobile GPU latency measured vs. predicted ones. The latency RMSE is 0.0571ms.

Searched MoGA Models

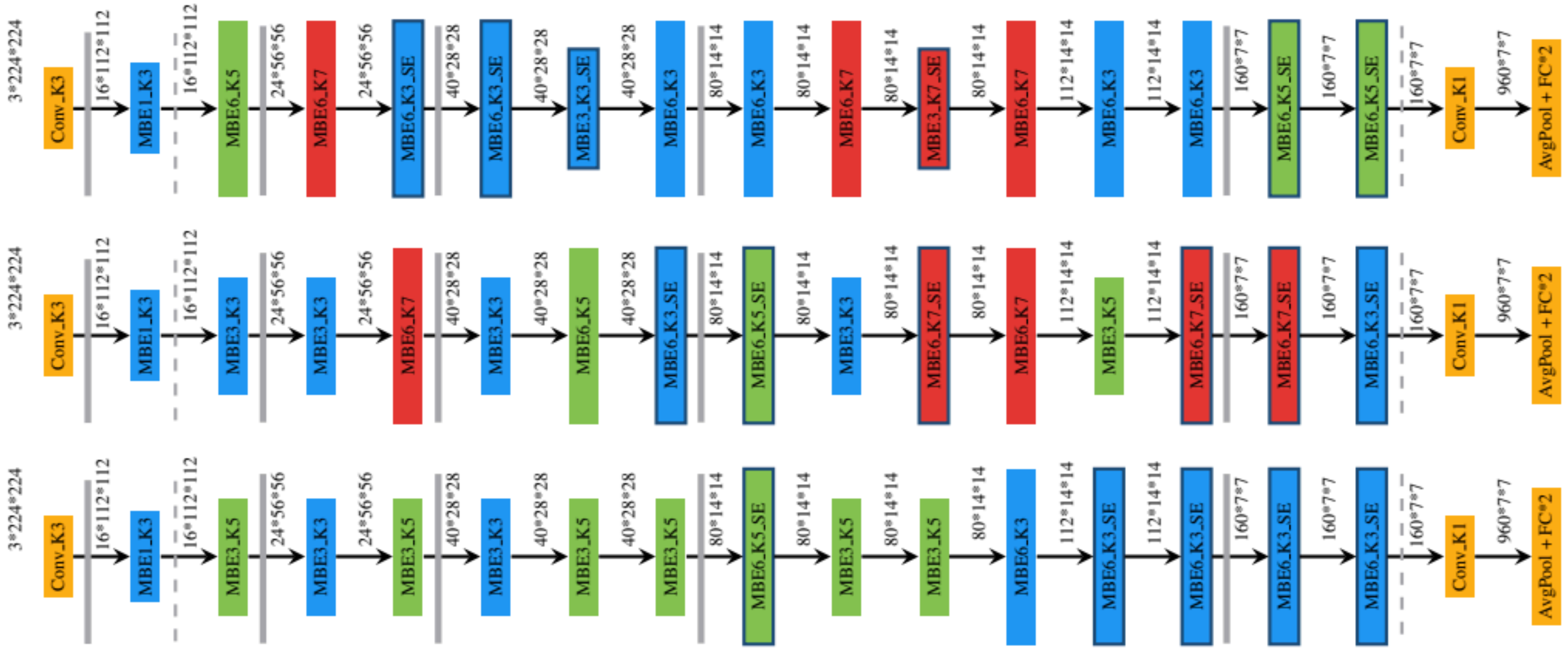


Fig. 4. The Architectures of MoGA-A, B, C, from top to bottom. Note Ex_Ky_SE means an expansion rate of x for its expansion layer and a kernel size of y for its depthwise convolution layer, SE for squeeze-and-excitation. Grey thick lines refer to downsampling points.

Comparison with SOTA Mobile Models

- MoGA-A with 75.9% accuracy with close mobile GPU latency to MobileNetV3 (75.0%)
- MoGA-C has faster speed on mobile GPU that MobileNetV3 with higher accuracy (75.3%)

Methods	$\times+$ (M)	P (M)	L_g^S (ms)	L_g^M (ms)	L_c (ms)	Top-1 (%)
MobileNetV2 [2]	300	3.4	$6.9^\dagger$	$7.0^\dagger$	78	72.0
MobileNetV3 [3]	219	5.4	$10.8^\star$	$9.5^\star$	66	$75.0^\star$
MnasNet -A1 [5]	312	3.9	-	-	78	75.2
MnasNet-A2 [5]	340	4.8	-	-	84	75.6
FBNet-B [9]	295	4.5	-	-	$23^\ddagger$	74.1
Proxyless-R [6]	$320^\dagger$	4.0	$7.3^\dagger$	$7.9^\dagger$	78	74.6
Proxyless GPU [6]	$465^\dagger$	7.1	$9.6^\dagger$	$9.8^\dagger$	124	75.1
Single-Path [10]	365	4.3	-	-	79	75.0
Once for All [27]	327	-	-	-	112^*	75.3
FairNAS-A [7]	388	4.6	$9.8^\dagger$	$9.7^\dagger$	104	75.3
MoGA-A (Ours)	304	5.1	11.8	11.1	101	**75.9**
MoGA-B (Ours)	248	5.5	10.3	10.0	81	75.5
MoGA-C (Ours)	221	5.4	9.6	8.8	71	75.3

Table 1. Comparison of mobile models on ImageNet. P: Number of parameters, L_g^S (L_g^M): SNPE (MACE) latency on mobile GPU, L_c: TFLite latency on CPU *: Our reimplementation. †: Based on its published code. ‡: Samsung Galaxy S8. *: Samsung Note 8.

Mobile GPU-awareness Analysis

- What do we learn:
    - Mobile CPU: Prefer fewer element-wise ops
    - Mobile GPU: Allow more percentages on element-wise ops

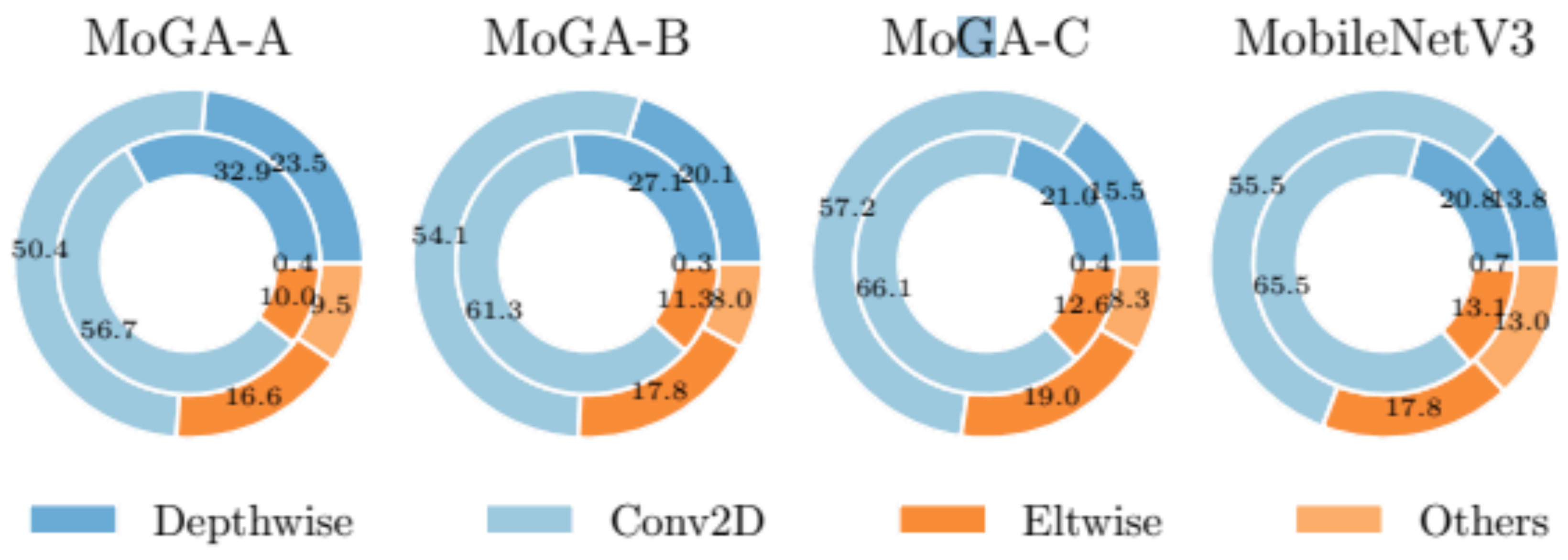
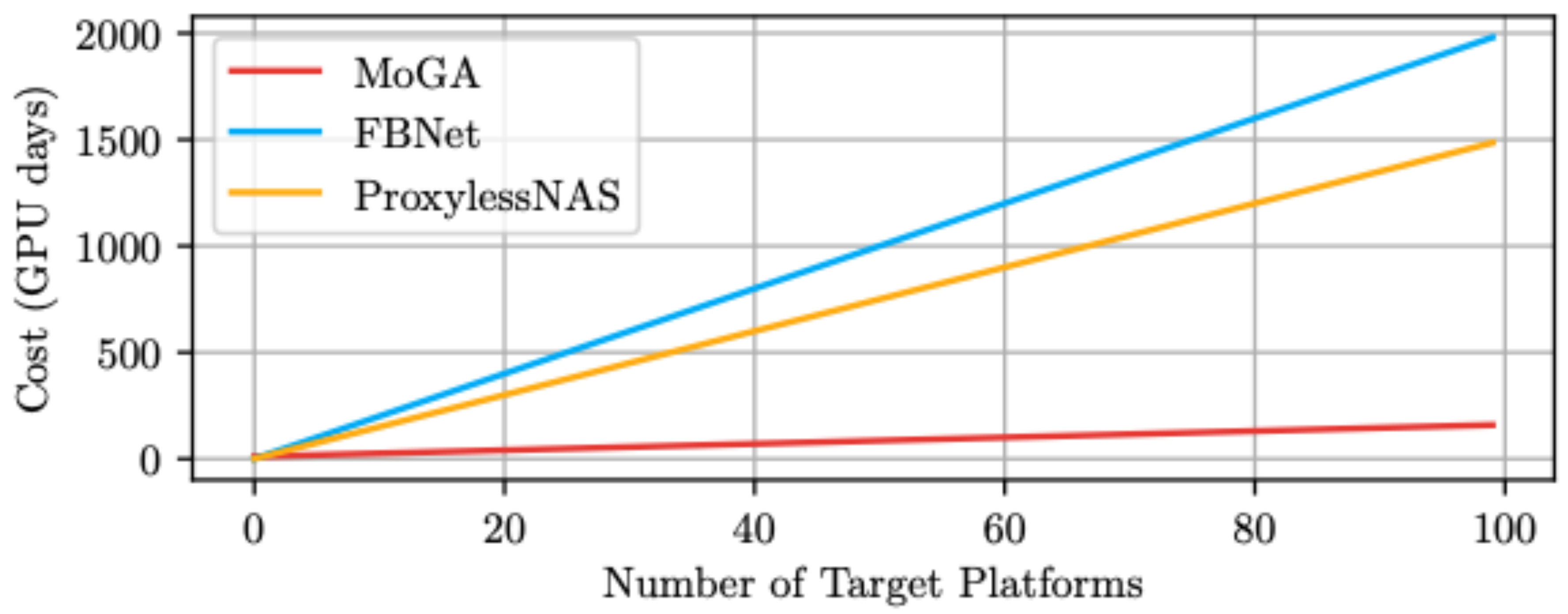


Fig. 1. Latency pie chart of MoGA-A, B, C and MobileNetV3 operations when run on mobile CPUs (inner circle with TFLite) vs. on mobile GPUs (outer circle with MACE).

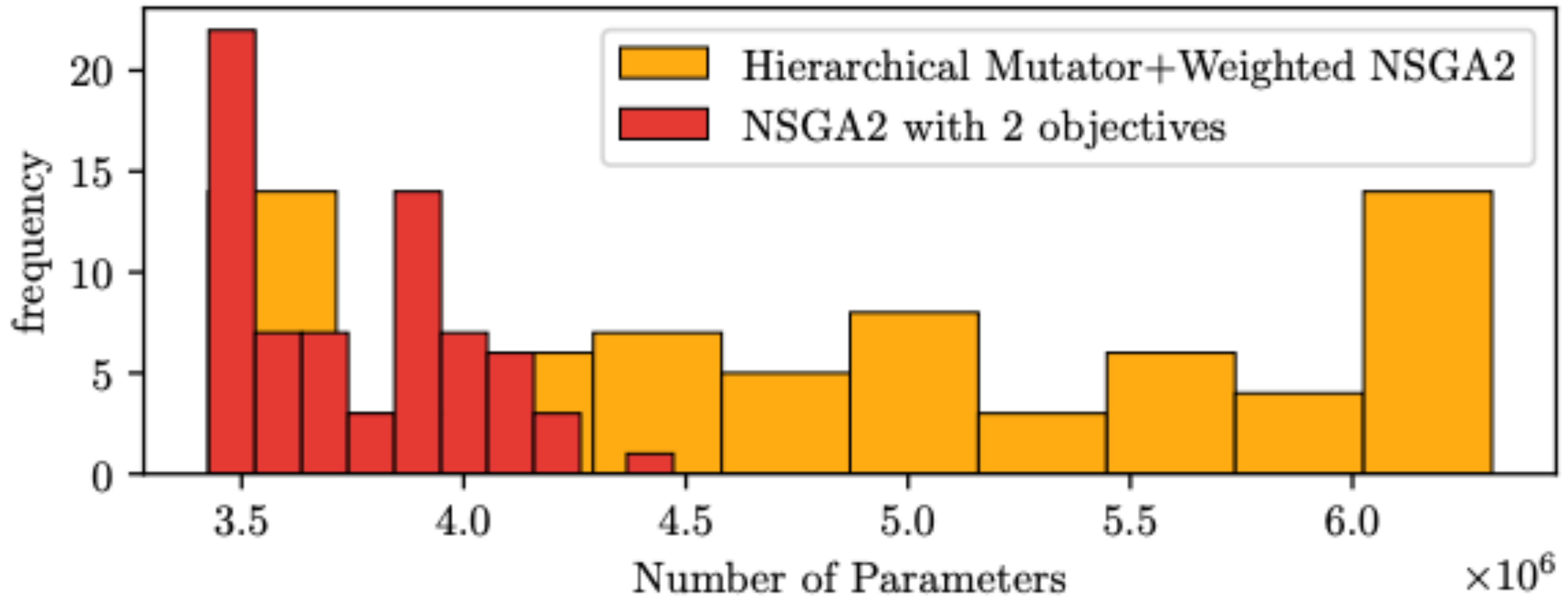
Search Cost Analysis

- 12 GPU days training & searching (200x less that MnasNet)
- For different target platforms,
  - Rebuild a latency lookup table
  - O(1) cost to run the search (the supernet is trained only once)



Why Three Objectives?

- Encouraging more number of parameters expands the searched model range, which is expected in mobile end



Ablation Study on Search Algorithms

- Search Algorithms
  - Weighted MoreMNAS (Chu et al. 2019), a variant of NSGA-II with three objectives
  - Vanilla weighted NSGA-II with three objectives
  - NSGA-II with two objectives

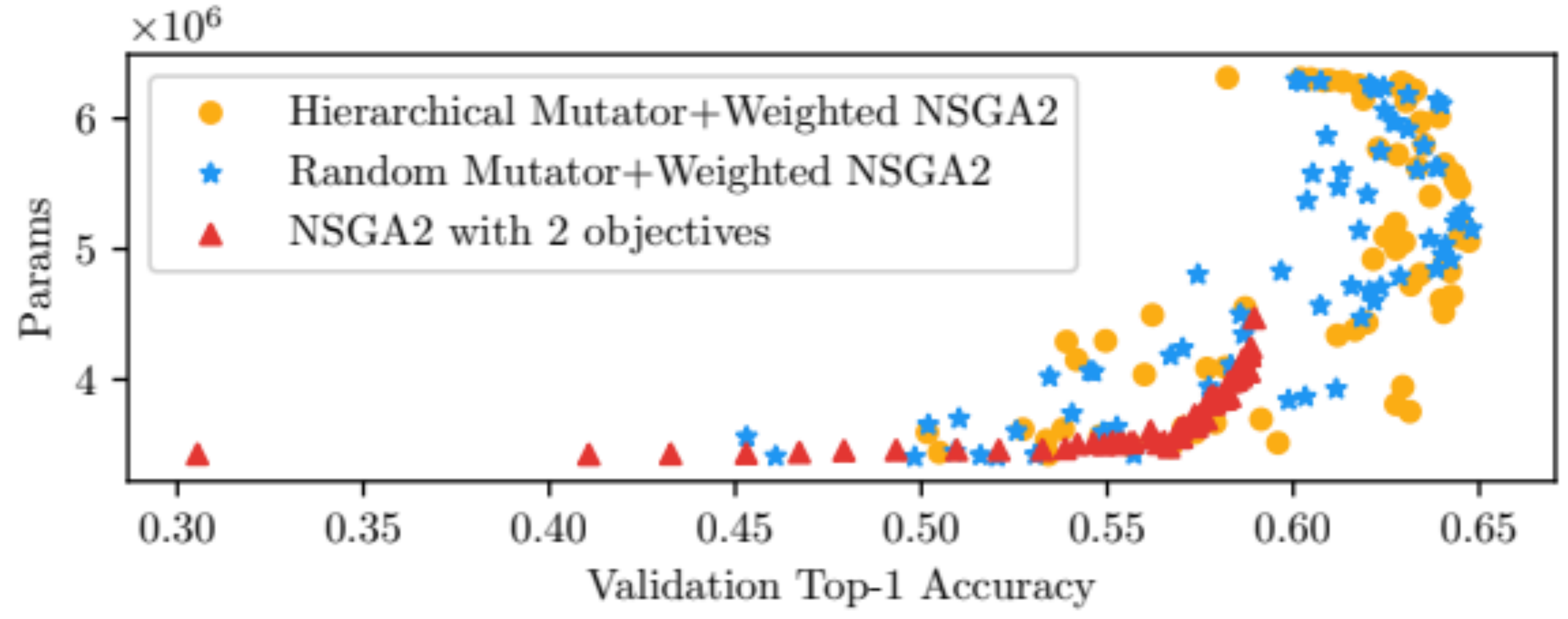


Fig. 5. Pareto Front of weighted NSGA-II with hierarchical mutator compared with that of a random mutator and of two objectives (accuracy, latency).

Conclusion

- Hardware-specific design is helpful
- Solving with MOP is necessary
- Three objectives expands the searched range
- One-shot supernet is less costly
- Mobile inference framework discrepancy could also be exploited

Thanks for watching!

If you still have some questions, please send emails to us.
(zhangbo11@xiaomi.com)