A. Supplementary Material

A.1. Toy experiment for share step size

We perform a toy experiment to compare individual step sizes for each candidate choice with a share step size for all candidate choices in one layer. The experiment is valid with the same settings except for step size design. The results are consistent with our observation that the range of activation and weights can be roughly the same, therefore, the share step size can be used to support the quantized supernet training.

Table 1. Toy experiment for step size settings. Individual denoted the second choice and share denotes the third choice in the main text. K3 is short for setting the kernel size of all blocks to 3.

	K3	K5	K7
Individual Step Size	74.78	75.39	75.40
Share Step Size	74.99	75.74	75.78

A.2. Proof of Bit Inheritance

Given a convolution layer in the K bit supernet, we denote s_K as the scale parameters of this layer, and $s_{K-1} = 2s_K$ as the doubled scale for the $K-1$ bit supernet. We use $\mathbf{w}$ and $N_{\mathbf{w}}$ to denote the weights of this layer which is inherited from K to $K-1$. Next, we show that the L_1 distance of $Q(\mathbf{w}, s_K)$ and $Q(\mathbf{w}, s_{K-1})$ is bounded by $N_{\mathbf{w}} \cdot s_K$. It means the initialized $Q(\mathbf{w}, s_{K-1})$ has a bounded distance with the well-trained $Q(\mathbf{w}, s_K)$.

For each w_i, we have:

$$\begin{aligned} Q(\mathbf{w_i}, s_K) &= \lfloor \text{clip}(\frac{\mathbf{w}_i}{|s_K|}, -2^{K-1}, 2^{K-1}-1) \rceil \times |s_K|, \\ Q(\mathbf{w_i}, s_{K-1}) &= \lfloor \text{clip}(\frac{\mathbf{w}_i}{|s_{K-1}|}, -2^{K-2}, 2^{K-2}-1) \rceil \times |s_{K-1}| \\ &= 2\lfloor \text{clip}(\frac{\mathbf{w}_i}{|2s_K|}, -2^{K-2}, 2^{K-2}-1) \rceil \times |s_K|. \end{aligned} \tag{1}$$

Based on this expression, we further get:

$$\begin{aligned} &|Q(\mathbf{w_i}, s_K) - Q(\mathbf{w_i}, s_{K-1})| = \\ &|\lfloor \text{clip}(\frac{\mathbf{w}_i}{|s_K|}, -2^{K-1}, 2^{K-1}-1) \rceil - \\ &2\lfloor \text{clip}(\frac{\mathbf{w}_i}{|2s_K|}, -2^{K-2}, 2^{K-2}-1) \rceil| \times |s_K|. \end{aligned} \tag{2}$$

For any $\mathbf{w_i}$ and s_K, we have:

$$\begin{aligned} &|\lfloor \text{clip}(\frac{\mathbf{w}_i}{|s_K|}, -2^{K-1}, 2^{K-1}-1) \rceil - \\ 2&\lfloor \text{clip}(\frac{\mathbf{w}_i}{|2s_K|}, -2^{K-2}, 2^{K-2}-1) \rceil| \leq 1. \end{aligned} \tag{3}$$

Thus,

$$|Q(\mathbf{w_i}, s_K) - Q(\mathbf{w_i}, s_{K-1})| \leq |s_K|, \tag{4}$$

And,

$$||Q(\mathbf{w}, s_K) - Q(\mathbf{w}, s_{K-1})||_1 \leq N_w \cdot |s_K|. \tag{5}$$

A.3. Training details

Dataset config: We evaluate our method on the ImageNet dataset. The training dataset is made up of 1.28 million images with resolution 224 × 224 belonging to 1000 classes and the validation set has 50k images. For ImageNet training, we use the typical random resized crop, randomly horizontal flipping and color jitter of $[32/255, 0, 0.5, 0]$ for data augmentation. During evaluation, we first determine the active image size s, and resize the image into $\lceil s/0.875 \rceil \times \lceil s/0.875 \rceil$ and center crop $s \times s$ image.

Quantization aware training: We reimplement LSQ [6] as our base quantization method. For ImageNet classification task, we start from a floating-point model and finetune the model for 150 epochs. The optimizer is SGD with Nesterov momentum 0.9 and weight decay 3e-5, and the label smoothing ratio is 0.1. The initial learning rate is 0.04 under the batch of 1024, with the cosine annealing schedule. The dropout rate is 0.1. In the finetuning stage like OQAT@25, we take the weights of the subnet from the supernet and finetune for 25epochs. The initial learning rate is 0.0016. In the supernet training, only the learning rate and the number of epochs are different. For object detection task on COCO dataset, we first train the floating-point models for 50 epochs, the learning rate starts from 0.16 with batch size 128, the learning rate is decayed by 0.1 at epoch 35 and epoch 45. Then for quantization-aware training, the learning rate starts from 0.016, and the rest is the same with the floating point models training pipeline. The Backbone except the first convolution is quantized in object detection task.

OQAT procedure: Combining the advantages of OFA [4] and BigNas [24], the overall OQAT procedure is divided into four steps as follow:

Step0: we train the 4 bit biggest models in the search space. It follows the typical quantization-aware training, we use the floating-point pre-trained model as initialization and finetuning for 150 epochs. The learning rate is 0.04 with a batch-size of 1024.

Step1: in the supernet training phase, the biggest model obtained in step 0 is used as initialization. The input resolution, kernel size, width, and depth are randomly sampled. This whole process takes 200 epochs. In one iteration, four models are sampled with the sandwich rule [23], which is the biggest subnet and the smallest subnet, and two random sampled subnets. The learning rate is 0.02 with a batch size of 1024.

Table 2. **MBV2 Search Space:** MBConv refers to inverted residual block which has a '1×1 pointwise - $k \times k$ depthwise- 1×1 pointwise' structure without SE module [11], MBConv-SE is the MBConv block with SE module. Channels mean the number of output channels in this stage. Depth means the number of blocks or layers in this stage. Expand ratio refers to the expansion ratio of input channels which controls the width of the depthwise convolution. Convolution layers in the first and last have no expansion ratio. Kernel size refers to the kernel size k of the depthwise convolution.

Stage	Operator	Resolution	Channels	Depth	Expand ratio	Kernel size
	Conv	128×128 - 224×224	32	1		3
1	MBConv	64×64 - 112×112	16	1	1	3
2	MBConv	64×64 - 112×112	24	2, 3, 4	3, 4, 6	3, 5, 7
3	MBConv	32×32 - 56×56	40	2, 3, 4	3, 4, 6	3, 5, 7
4	MBConv	16×16 - 28×28	80	2, 3, 4	3, 4, 6	3, 5, 7
5	MBConv	8×8 - 14×14	96	2, 3, 4	3, 4, 6	3, 5, 7
6	MBConv	8×8 - 14×14	192	2, 3, 4	3, 4, 6	3, 5, 7
7	MBConv	4×4 - 7×7	320	1	3, 4, 6	3, 5, 7
	Conv	4×4 - 7×7	1280	1		1

Table 3. **MBV3 Search Space:** MBConv refers to inverted residual block which has a '1×1 pointwise - $k \times k$ depthwise- 1×1 pointwise' structure without SE module [11], MBConv-SE is the MBConv block with SE module. Channels mean the number of output channels in this stage. Depth means the number of blocks in this stage. Expand ratio refers to the expansion ratio of input channels which controls the width of the depthwise convolution. Convolution layers in the first and last have no expansion ratio. Kernel size refers to the kernel size k of the depthwise convolution.

Stage	Operator	Resolution	Channels	Depth	Expand ratio	Kernel size
	Conv	128×128 - 224×224	16	1		3
1	MBConv	64×64 - 112×112	16	1	1	3
2	MBConv	64×64 - 112×112	24	2, 3, 4	3, 4, 6	3, 5, 7
3	MBConv-SE	32×32 - 56×56	40	2, 3, 4	3, 4, 6	3, 5, 7
4	MBConv	16×16 - 28×28	80	2, 3, 4	3, 4, 6	3, 5, 7
5	MBConv-SE	8×8 - 14×14	112	2, 3, 4	3, 4, 6	3, 5, 7
6	MBConv-SE	8×8 - 14×14	160	2, 3, 4	3, 4, 6	3, 5, 7
	Conv	4×4 - 7×7	960	1		1
	Conv	1×1	1280	1		1

With bit inheritance, the training of the 3/2 bit supernet is simplified. And the training time is reduced.

Step2: In the 3-bit supernet, we use the 4-bit supernet obtained in the Architecture Shrinking Step1 part2 as initialization. We directly random sample input resolution, kernel, width, and depth. four models are sampled, which is the biggest subnet and the smallest subnet, and two random sampled subnets for one update. We only use 25 epochs and the learning rate is 0.0016.

Step3: In the 2-bit supernet, we use the 3-bit supernet obtained in Step2 as initialization. We also directly random sample resolution, kernel, width, and depth. Four models are sampled, which is the biggest subnet and the smallest subnet, and two random sampled subnets for one update. We only use 120 epochs and the learning rate is 0.0256.

OQAT subnet finetuning: Our OQAT performance can be further improved by finetuning the subnet weights sliced from the OQAT supernet as suggested by OFA [4]. The accuracy of the subnet is already higher than training from scratch. In default, the subnets are finetuned for 25epochs. The initial learning rate is 0.0016 with a batch size of 1024, with the cosine annealing schedule.

Knowledge distillation: The knowledge distillation(KD) used in our experiment is the traditional loss(KD) proposed in [9]. The student's logits and teacher's logits are used to calculating the cross-entropy loss. The temperate is 1 and the kd loss weight is 1.

Search Space: The details of search space is shown in Table 2 and Table 3. And we also compare with other quantization-aware NAS methods in Table 4. Quantization algorithm, based network architecture, bit width search space, and retrain or not are listed.

Architecture search of quantized supernet. We directly evaluate the sampled subnets from the supernet without further retraining. It's worth mentioning that we use the predictive accuracy on 10K validation images sampled from *trainset* to measure the subnets in the search procedure. Furthermore, we exploit a coarse-to-fine architecture selection

Table 4. **The details of quantization-aware NAS:** named SPOS [7], BMobi [17], BATS [2], APQ [21] are given, including network architecture, search space, bit-width and quantization algorithm PACT [5], Bireal [16], Xnor-net++ [3], HAQ [20], LSQ [6]. Group MobileNet denotes the MobileNet with group convolution in place of depthwise convoluton.

	SPOS	BMobi	BATS	APQ	OQAT
Quantization Algorithm	PACT	Bireal	Xnor-net++	HAQ	LSQ
Network Architecture	ResNet	Group MobileNet	Group Darts	MobileNetV2	MobileNetV2, MobileNetV3
Bit Width	{1, 2, 3, 4}	{1}	{1}	{4, 6, 8}	{2, 3, 4}
Search Space	width, bit-width	group number	operation, connection	width, depth, kernel size, bit-width	width, depth, kernel size, resolution
Retrain	✓	✓	✓	✓	×

Table 5. ImageNet performance under 4, 3, 2 bit. OQAT-4bit-M and OQAT-4bit-L denote medium and large model size in the 4-bit OQATNets family respectively. OQAT means we take weights from the supernet directly and OQAT@25 means we take weights from the supernet and finetune for 25 epochs. LSQ* results for OQAT models means we train these models from scratch individually. W/A denotes the bit-width for both weights and activation. BitOPs is calculated by [22, 14].

Models	Method	Bit (W / A)	BitOPs (G)	Top-1 Acc.(%)
Efficient-B0	QKD	4	6.78	73.1
OQAT-4bit-L	LSQ*	4	4.67	73.3
OQAT-4bit-L	OQAT	4	4.67	73.4
OQAT-4bit-L	OQAT@25	4	4.67	**74.1**
ResNet-18	LSQ / LSQ*	4	34.6 / 34.6	71.1 / 70.9
MobileNetV2	LSQ* / SAT	4	5.44	71.3 / 71.1
MbV3-L (1.0x)	LSQ*	4	3.84	71.7
OQAT-4bit-M	OQAT@25	4	3.0	**72.3**
ResNet-18	LSQ / APOT	3	22.8 / 19.1	70.6 / 69.9
Efficient-B0	QKD	3	4.16	69.2
OQAT-3bit-L	OQAT@25	3	3.07	**71.3**
MobileNetV2	LSQ* / QKD	3	3.39 / 3.39	68.2 / 62.6
OQAT-3bit-L	LSQ*	3	3.07	**70.5**
MbV3-L 1.0x	LSQ*	3	2.43	67.5
OQAT-3bit-M	LSQ*	3	1.92	**67.2**
OQAT-3bit-M	OQAT@25	3	1.92	**68.3**
Efficient-B0	QKD / LSQ+	2	2.30	50.0 / 49.1
MobileNetV2	LSQ* / QKD	2	1.92	55.7 / 45.7
OQAT-2bit-L	OQAT@25	2	1.60	**64.0**
OQAT-2bit-S	OQAT@25	2	0.89	**57.7**

procedure, similar to [24]. We first randomly sample 10K candidate architectures from the supernet with the FLOPs of the corresponding floating-point models ranging from 50M to 300M (2K in every 50M interval). After obtaining the good skeletons (input resolution, depth, width) in the pareto front of the first 10K models, we randomly perturb the kernel sizes to further search for better architectures.

A.4. Further comparison with existing architectures

We also verify our searched architecture under downstream task for Object Detection.

We evaluate several architectures with strong quantization methods including LSQ [6], LSQ+ [1], APOT [14], QKD[13], SAT [12], and LSQ* which is the LSQ implemented by us on different models to construct strong baselines. The result of 4 bit ResNet-18@LSQ* validates that our implementation is comparable.

Our OQAT benefits from joint quantization and network architecture search, as well as the bit inheritance for lower bits. As shown in Table 5, our OQATNets outperforms multiple quantization methods on models like MobileNetV2 [18], EfficientNet-B0 [19] and MbV3 [10] under all bit-widths we implements. **4 bit:** OQAT4bit-L has 1% accuracy gain higher than Efficient-B0@QKD. OQAT4bit-M outperforms ResNet-18@LSQ with 10% of

Table 6. **3-bit** Quantization-Aware Training results on the COCO dataset, the training pipeline are the same, except for the backbone architectures.

	FLOPs	AP_{FP}	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
ResNet18	1800M	**33.8**	32.4	49.2	33.9	16.4	33.0	45.3
MobileNetV3	219M	33.5	30.4	46.2	32.1	16.7	32.7	42.3
OQAT-3bit-M	222M	33.2	30.8	47.2	32.7	**17.4**	32.5	41.8

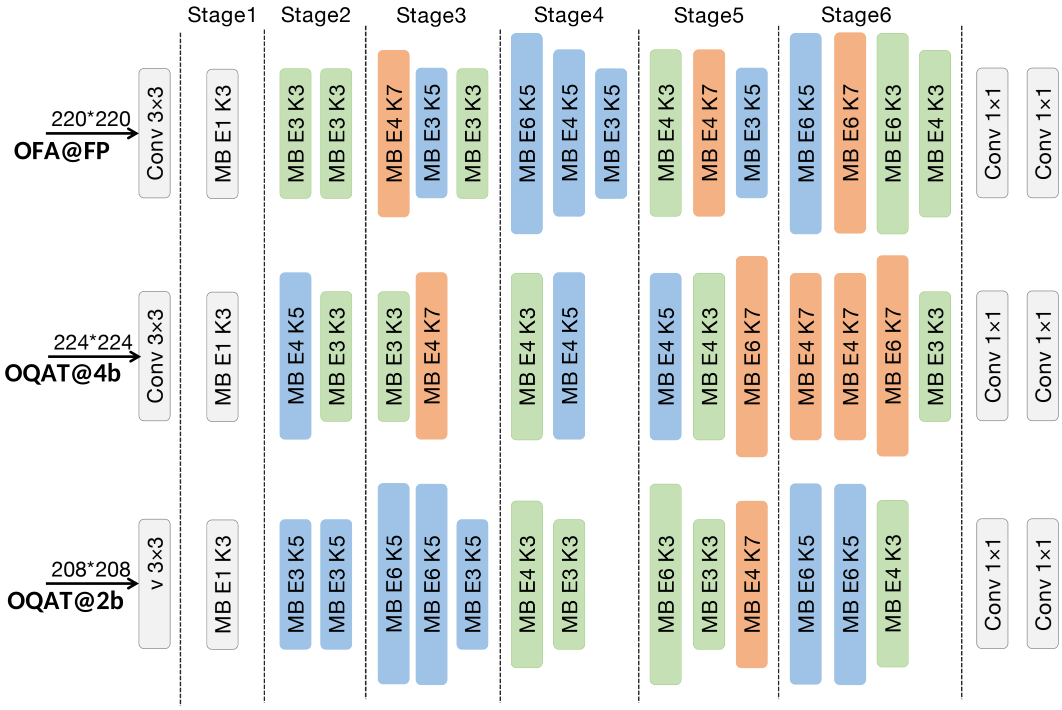

Figure 1. **Architecture visualization:** of OFANet and our searched OQATNets. 'MB E3 K3' indicates 'mobile block with expansion ratio 3, kernel size 3x3'. From top to bottom, there are FP OFANet, 4-bit OQATNet and 2-bit OQATNet. There are under similar computation cost, around 220M FP FLOPs.

its FLOPs. **3 bit:** Our OQAT3bit-L can also match the accuracy of 3 bit ResNet-18@LSQ with 13% FLOPs and 3 bit Efficient-B0@QKD with 74% FLOPs. **2 bit:** Our OQAT2bit-L requires less FLOPs but achieves significantly higher Top-1 accuracy (64.0%) when compared with EfficientNet-B0@QKD (50.0%) and MobileNetV2@LSQ* (55.7%). The results verify that the joint design of quantization and NAS results in more quantization-friendly compact models.

Train-from-scratch: we present some training-from-scratch results for OQAT-4bit-L, OQAT-3bit-L and OQAT-3bit-M to compare with existing models under the same training routine. The OQAT-4bit-L@LSQ* has similar accuracy while has much fewer BitOPs, which verifies the superiority of our searched architecture. Similarly, the result of OQAT-3bit-L@LSQ* has 2% accuracy gain with MobileNetV2@LSQ* with 0.3G fewer BitOPs. The accuracy of OQAT-3bit-M@LSQ* is roughly the same with MobileNetV3@LSQ*, while we have 0.5G fewer BitOPs. Although the supernet benefits from distillation, the train-from-scratch results verify that we do search for a better quantization-aware friendly models compared with existing efficient architectures MobileNetV2/MobileNetV3/EfficientNetB0.

A.5. Further verification on Object Detection Task

To further verify our searched architecture on different tasks, we perform quantization on RetinaNet [15]. ResNet18 [8], MobilenNetV3 [10] are used to fairly compared with OQAT-3bit-M. OQAT-3bit-M has 0.4% gain in terms of AP than MobileNetV3. OQAT-3bit-M has 1% gain in AP_s than ResNet18.

A.6. Visualization of searched architecture under different bits.

In Figure 1, the searched architeture with similar FLOPs under different bit-widths are presented. The searched quantization models are clearly wider and shallower compared with floating-point models. It is consistent with the analysis presneted in the Section 5.5.

References

[1] Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. Lsq+: Improving low-bit quantization through learnable offsets and better initialization. *arXiv preprint arXiv:2004.09576*, 2020. 3

[2] Adrian Bulat, Brais Martinez, and Georgios Tzimiropoulos. Bats: Binary architecture search. *arXiv preprint arXiv:2003.01711*, 2020. 3

[3] Adrian Bulat and Georgios Tzimiropoulos. Xnor-net++: Improved binary neural networks. *arXiv preprint arXiv:1909.13863*, 2019. 3

[4] Han Cai, Chuang Gan, and Song Han. Once for all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019. 1, 2

[5] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018. 3

[6] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019. 1, 3

[7] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. Single path one-shot neural architecture search with uniform sampling. *arXiv preprint arXiv:1904.00420*, 2019. 3

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[10] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1314–1324, 2019. 3, 4

[11] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 2

[12] Qing Jin, Linjie Yang, and Zhenyu Liao. Towards efficient training for neural network quantization. *arXiv preprint arXiv:1912.10207*, 2019. 3

[13] Jangho Kim, Yash Bhalgat, Jinwon Lee, Chirag Patel, and Nojun Kwak. Qkd: Quantization-aware knowledge distillation. *arXiv preprint arXiv:1911.12491*, 2019. 3

[14] Yuhang Li, Xin Dong, and Wei Wang. Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. *arXiv preprint arXiv:1909.13144*, 2019. 3

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 4

[16] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *Proceedings of the European conference on computer vision (ECCV)*, pages 722–737, 2018. 3

[17] Hai Phan, Zechun Liu, Dang Huynh, Marios Savvides, Kwang-Ting Cheng, and Zhiqiang Shen. Binarizing mobilenet via evolution-based searching. *arXiv preprint arXiv:2005.06305*, 2020. 3

[18] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 3

[19] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019. 3

[20] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8612–8620, 2019. 3

[21] Tianzhe Wang, Kuan Wang, Han Cai, Ji Lin, Zhijian Liu, Hanrui Wang, Yujun Lin, and Song Han. Apq: Joint search for network architecture, pruning and quantization policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2078–2087, 2020. 3

[22] Bichen Wu, Yanghan Wang, Peizhao Zhang, Yuandong Tian, Peter Vajda, and Kurt Keutzer. Mixed precision quantization of convnets via differentiable neural architecture search. *arXiv preprint arXiv:1812.00090*, 2018. 3

[23] Jiahui Yu and Thomas S Huang. Universally slimmable networks and improved training techniques. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1803–1811, 2019. 1

[24] Jiahui Yu, Pengchong Jin, Hanxiao Liu, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, Thomas Huang, Xiaodan Song, Ruoming Pang, and Quoc Le. Bignas: Scaling up neural architecture search with big single-stage models. *arXiv preprint arXiv:2003.11142*, 2020. 1, 3