6. Appendix

6.1. Uniform-sum Compression Ratio Sampling

We share the details of our sampling algorithm here:

Algorithm 1 Uniform-Sum Compression Ratio Sampling

Input: Number of layers n, predefined sum range $[s_{\min}, s_{\max}]$, upper bound u
Output: compression ratios $\{r_0, r_1, \ldots, r_n\}$ such that $\sum r_i = S$
$S_{\min} \leftarrow s_{\min} \times 10$, $S_{\max} \leftarrow s_{\max} \times 10$, $U \leftarrow u \times 10$
Sample a target sum of compression ratios:
    $S \sim \text{Uniform}(S_{\min}, S_{\max})$
repeat
    Sample compression ratios from a Dirichlet distribution:
        $(r_0, r_1, \ldots, r_n) \sim \text{Dirichlet}(\alpha)$
    Scale sampled values to ensure their sum equals S:
        $r_i \leftarrow r_i \cdot S$, for all i
until $r_i \leq U$ for all i
Apply Largest Remainder Method (LRM) for rounding:
Round down each element: $R \leftarrow \lfloor r_i \rfloor$
Compute remaining difference: diff $\leftarrow S - \sum R$
if diff > 0 **then**
    Compute fractional remainders: remainder $\leftarrow r_i - R$
    Sort indices by largest remainder in descending order
    for $i = 1$ to diff **do**
        Increment R at the index with the highest remainder
    end for
end if
for $i = 0$ to n **do**
    $r_i \leftarrow R_i/10$
end for
Return $\{r_0, r_1, \ldots, r_n\}$

For the Swin-Tiny backbone that we use in the experiments, the number of blocks is 6, and the upper bound for compression ratios is 0.8.

We aim for the relative computational cost to fall within the range $[0.65, 0.95]$. Given that MixA-Q reduces computations by 50%, the corresponding range for the sum of compression ratios is computed as:

$[(1 - 0.95) \times 6 \times 2, (1 - 0.65) \times 6 \times 2] = [0.6, 4.2]$

Note that during sampling, we assume that all blocks have an equal number of BOPs. However, in our experiments, the reported relative computational cost is recalculated based on the actual distribution of BOPs across different blocks.

In Tab. 6 we compare the SAQA using uniform-sum sampling and naive uniform sampling. It can be observed that SAQA with uniform-sum sampling constantly achieves higher mAP at different averaged activation bits.

	Act Bit	mAP
SAQA uni.sum	3.36*	43.4
	3.23*	43.2
	3*	42.4
SAQA	3.36*	43.0
	3.23*	42.8
	3*	41.9

Table 6. mAP performance comparison of SAQA using uniform-sum sampling and naive uniform sampling. * indicates that the bit width is a computationally weighted average.

In terms of sampling efficiency, as shown in Fig. 8, our uniform-sum sampling is able to find configurations for MixA-Q that is better than the PTQ baseline after only two generations of the evolutionary search, while naive sampling can't. This means that our uniform-sum sampling helps the search process to converge faster and thus saves searching time.

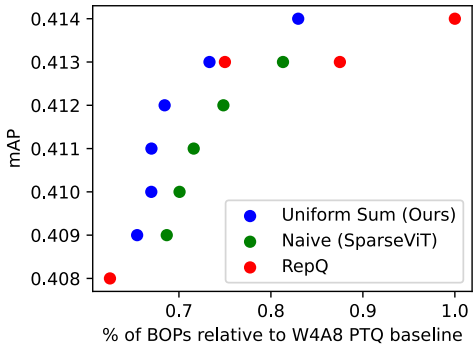

Figure 8. Pareto front after two generations of evolutionary search using different sampling methods.

6.2. Calculation of Equivalent Activation Bits

In this section we explain how the equivalent activation bits (marked with *) in our experiment part are calculated for MixA-Q and SparseViT. Given the GMACs (Giga Multiply-Accumulate operations) of the Swin-Tiny backbone:

$$\text{GMACs} = [13.3, 13.56, 14.16, 14.16, 14.16, 14.02] \quad (1)$$

$$C = 3 \quad (2)$$

GMACs contains the number of GMACs for each pair of consecutive Swin blocks and the C is the number of GMACs of for downsampling layers whose computations can't be saved by MixA-Q or SparseViT.

In MixA-Q, 50% computations of the compressed windows are saved while 100% computations of the pruned windows are saved in SparseViT. Thus, for a given com-

pression or pruning ratio configuration RC and the activation bit of the baseline model ActBit_{base} the equivalent activation bits ActBit_{eq} is calculated as:

$$\text{ratios} = \left(1 - \frac{r}{2}\right), \quad \forall r \in RC, \text{if method is MixA-Q} \tag{3}$$

$$\text{ratios} = 1 - r, \quad \forall r \in RC, \text{if method is SparseViT} \tag{4}$$

$$\text{ActBit}_{eq} = \text{ActBit}_{base} \times \frac{\sum_i (\text{ratios}_i \cdot \text{GMACs}_i) + C}{\sum_i \text{GMACs}_i + C} \tag{5}$$