1. Implementation Details

This section primarily introduces the selection of distillation layers and the implementation of the differentiable counter.

1.1. Fine-tuning Knowledge Distillation Layers

As mentioned in the previous section 5.1, we select certain outputs of the intermediate layers for distillation during the training and fine-tuning process. Specifically, we choose the output of each block of the models as knowledge distillation layers, taking into account both the stability of the model's convergence and its final compression performance during our experiments.

1.2. Differentiable Counter

In the process of implementing the differentiable counter, the weights $\hat{\boldsymbol{w}}$ are first sorted, and then binary search is used to determine the position of the query value x in the sorted weights. By using relaxation, the forward propagation value can be obtained. In the process of calculating gradients during backward propagation, it is important to take note of the accumulation of associated gradients. In practice, weight sampling is used to count the frequencies of values when the number of parameters in a layer exceeds a certain amount (such as 2 to the power of 14) due to the relatively slow traversal of counts. We have observed that the utilization of sampling leads to a substantial improvement in computational efficiency while maintaining a satisfactory level of accuracy. To this end, we offer two implementations: one in C++ and the other in CUDA [2].

2. Additional Experimental Results

In this section, we present additional experimental results and analyses that were not included in the main text. These results include the analysis of compression ratio allocations for layers in more networks, ablation experiments on encoding algorithms, calibration dataset size, and weight fine-tuning, as well as experiments on a novel network architecture.

2.1. Compression Ratio Allocation for Layers

In this section, we will present more distributions of bit width and sparsity ratio for the compressed models, which were not shown in the experiment section. We have selected one representative model each from MobileNet, RegNet, and MNasNet. Since these models exhibit varying levels of accuracy at different compression ratios, we have chosen the highest compression ratios possible while maintaining an acceptable level of accuracy.

MobileNetV2. Figure 1 depicts the distribution of bit width and sparsity ratio for MobileNetV2 at a $10\times$ compression ratio. Our observations indicate that the earlier layers of the network generally have higher bit width values, while the later layers have lower bit width values. Additionally, sparsity is rarely observed in the earlier layers, but alternates between high and low ratios in the later layers. Remarkably, the final layer, which is a classifier, exhibits the highest sparsity ratio across the entire network, which contrasts sharply with the characteristics of other models.

Entropy Loss	Huffman Coding	Range Coding
4517.31KB	4588.33KB	4527.30KB
3719.25KB	3799.03KB	3721.24KB
3033.46KB	3130.71KB	3040.09KB
2496.27KB	2669.75KB	2508.25KB
2277.14KB	2500.24KB	2288.99KB

Table 1. Effect of different coding methods on ResNet-18.

RegNet-600m. Figure 2 displays the distribution of bit width and sparsity ratio for RegNet-600m at a $15\times$ compression ratio. Our observations show that the compression ratio is more pronounced in the later layers of RegNet-600m, with lower bit width values and higher sparsity ratios. Conversely, the earlier layers exhibit lower compression ratios with lower sparsity ratios and higher bit width values. Meanwhile, the compression ratio of the classifier layer remains relatively low.

MNasNet. Figure 3 presents the result of MNasNet at a compression ratio of $12\times$. It is noteworthy that the layers with lower bit width values are located towards the end of the network, whereas the layers with higher bit width values are situated towards the front. This discrepancy is more pronounced compared to other networks.

2.2. Additional Ablation Study

In this section, we will provide some additional ablation studies, including the choice of lossless compression methods, the impact of calibrating dataset length, and the effect of weight fine-tuning.

Choice of lossless compression. We demonstrate the impact of using different encoding algorithms on the compression results, mainly comparing Huffman coding with range coding. Table 1 presents the theoretical compression values calculated using the entropy regularization term under different target compression ratios for ResNet-18, the compressed model sizes after using Huffman coding, and the compressed model size after using range coding. It is apparent that the results of range coding are closer to the theoretical compression values, especially when the compression ratios are high. In such scenarios, the difference between

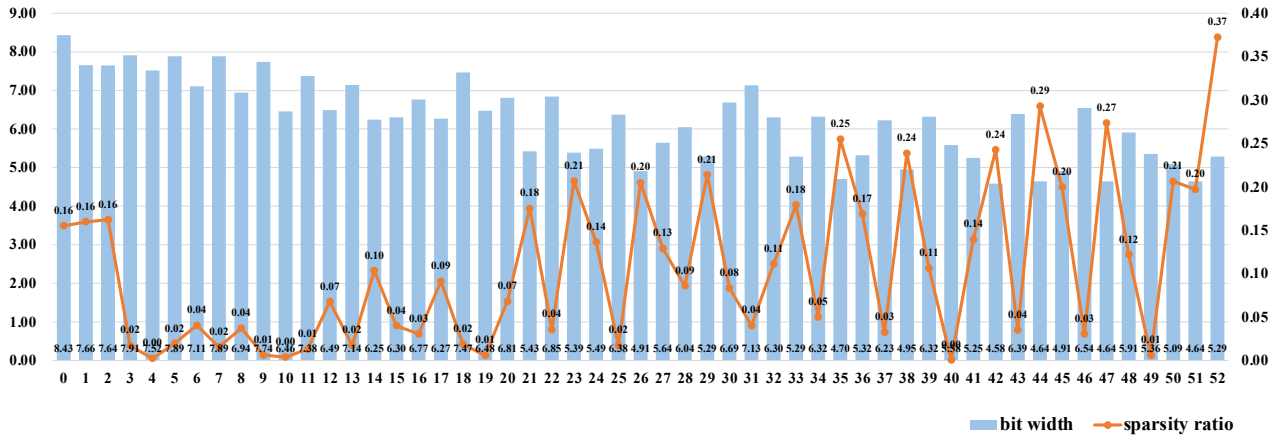

Figure 1. The distribution of bit width and sparsity ratio of MobileNetV2 at a compression ratio of 10×.

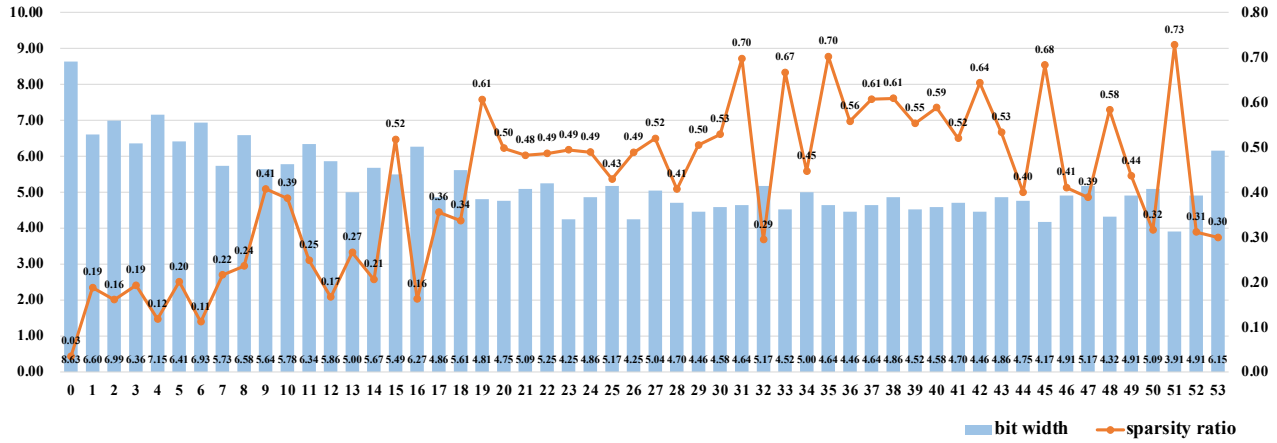

Figure 2. The distribution of bit width and sparsity ratio of RegNet-600m at a compression ratio of 15×.

the two coding methods is significant, and the advantages of range coding are more pronounced.

The reason lies in the fact that Huffman coding is a symbol coding method that uses only integer bits for encoding, which imposes certain limitations on the compressed data. In contrast, range coding does not suffer from such limitations and, as a result, offers more potential for achieving higher compression ratios. This is because range coding uses fractional bits for encoding, allowing for a more precise representation of the data and reducing the number of bits needed to represent the same information.

Choice of calibration dataset size. In Table 2, we observe that in the ImageNet-1k classification task, when the number of images in the calibration dataset is less than 768, the model's accuracy drop is more severe. However, when the number of images is greater than 768, the model's performance does not show significant improvement. Nevertheless, using more data for calibration would have negative impacts on calibration time and data acquisition. Therefore, we have opted to use 1000 images as the calibration dataset, striking a balance between different aspects.

Length	2000	1000	768	512	256
Accuracy/%	70.78	70.79	70.75	69.79	67.93

Table 2. The results for ResNet-18 at 12× compression ratio with different calibration sizes.

Effect of weight fine-tuning. We evaluate the performance of our unified compression method without weight fine-tuning, as shown in Table Table 3. Our results demonstrate that our method can maintain a compression ratio of 10×

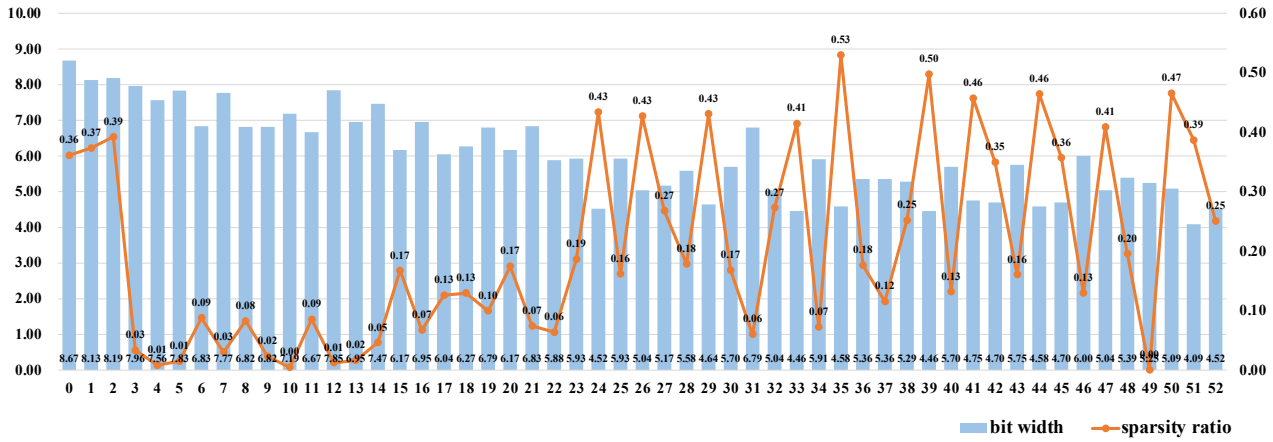

Figure 3. The distribution of bit width and sparsity ratio of MNasNet at a compression ratio of 12×.

Model	CR	Top1-acc
ResNet-18	10.05×	70.47%
ResNet-50	10.04×	76.22%
MobileNetV2	9.864×	71.46%
RegNet-3200m	9.935×	78.00%
RegNet-600m	9.903×	72.82%
MNasNet	9.980×	75.78%

Table 3. The results without weight fine-tuning.

Target CR	CR	Top-1 acc
-	1.00×	85.09%
6×	6.67×	84.97%
8×	8.05×	84.83%
10×	10.05×	84.36%
12×	12.01×	83.44%
14×	14.02×	81.86%

Table 4. ViT compression results.

with minimal accuracy drop, solely relying on the unified transformation $\mathcal{T}(\cdot)$. This approach is still superior to existing state-of-the-art methods, highlighting the rationality and robustness of our proposed method.

2.3. Performance on Vision Transformer

We also validated our method on the popular Vision Transformer network [1]. Specifically, we conduct classification experiments on the ViT-base-patch16 using the same settings as the previous experiments. However, considering the computational complexity of the model and the need for efficiency, we set the batch size to 16 and train for 3 epochs, with 300 iterations for training transformation parameters per epoch and 1000 iterations for fine-tuning weights. As shown in Table Table 4, our method still achieves good performance on this network.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[2] NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek. Cuda, release: 10.2.89, 2020.