A. Implementation Details

For vision models including both CNNs and Transformers, we use only 1K images randomly sampled from the whole training dataset (i.e., *train* dataset of ImageNet [8]). During the optimization, We use Adam [3] optimizer whose initial learning rates for scaling factors α_i, bit-code $\mathbf{b}_i$, and the step size of activations are 0.00035, 0.001, and 0.00004, respectively. They are then decreased to zero by Cosine annealing [5], except the learning rate for bit-code. We set the batch size to 32 for Convolutional models while setting it to 12 for Transformer models, except ViT-L (set to 2) due to a memory issue. The regularization parameter λ used in Eq. (9) of the manuscript is set to 0.7 for MobileNetV2 [9] and MnasNet [11], and to 0.01 for other models. We have obtained pre-trained models from public repositories[1,2,3].

For natural language processing (NLP) models, we use pre-trained models of BERT [2] and DistilBERT [10] from huggingface[4]. To evaluate the performance, we use a subset of the training dataset by random sampling: 3,377 samples (3.8% of training dataset) for SQuAD v1.1 [7], and 10K samples (2.5% of the training dataset) for MNLI [12]. However, the whole dataset is used for MRPC [12]. Moreover, we set the batch size to 12 and 8 for BERT and DistilBERT, respectively. All other hyperparameters are the same as in vision models.

For all experiments in the paper, we use NVIDIA Tesla V100 with 32GB memory. The training time and peak memory usage for each model are shown in Table A1, A2, and A3. Note that the training cost may vary in each run but is insignificant. Bit-width also has a negligible influence on the cost.

Table A1. Training Cost for CNNs

Model (W2A4)	ResNet-18	ResNet-50	MobileNetV2
Time (hrs)	0.57	3.07	1.02
Mem (GB)	2.26	8.67	4.32

Model (W2A4)	RegNetX -600MF	RegNetX -3.2GF	MnasNet
Time (hrs)	1.10	2.13	1.80
Mem (GB)	2.95	8.22	7.32

Table A2. Training Cost for Vision Transformer Models

Model (W2A8)	ViT-B	ViT-L	DeiT-S	DeiT-B
Time (hrs)	4.87	4.30	0.80	1.70
Memory (GB)	25.76	22.59	3.56	8.93

[1]https://github.com/yhhhli/BRECQ
[2]https://github.com/google-research/vision_transformer
[3]https://github.com/facebookresearch/deit
[4]https://github.com/huggingface/transformers; v4.10.0

Table A3. Training Cost for Language Transformer Models

Model (W2A8)	BERT		
	SQuAD v1.1	MRPC	MNLI
Time (hrs)	3.65	1.53	1.95
Mem (GB)	17.90	7.98	7.98

Model (W2A8)	DistilBERT		
	SQuAD v1.1	MRPC	MNLI
Time (hrs)	1.72	0.78	0.80
Mem (GB)	9.53	5.33	5.33

Table A4. The Evaluation on Vision Transformers

W2A8	ViT-B	ViT-L	DeiT-S	DeiT-B
Mr.BiQ	75.46$\pm$0.11	75.86$\pm$0.12	73.15$\pm$0.16	78.97$\pm$0.07
Mr.BiQ-U†	73.35$\pm$2.54	74.93$\pm$0.78	73.09$\pm$0.11	78.96$\pm$0.07
BRECQ	71.52$\pm$2.76	72.45$\pm$1.04	68.92$\pm$0.15	76.91$\pm$0.13
BRECQ-B‡	70.20$\pm$2.62	71.97$\pm$1.58	69.00$\pm$0.15	76.54$\pm$0.13

† Mr.BiQ-Uniform.
‡ It jointly optimizes the step size in addition to bit-code.

B. Comparison with Integer-based Optimization

We can regard estimating the optimal combination of α_i's and $\mathbf{b}_i$'s as a regression problem. In this perspective, optimizing the step size and bit-code acts as a feature in linear regression. Accordingly, prior works optimizing only a single feature correspond to solving a univariate linear regression while Mr.BiQ, which optimizes both features, corresponds to solving a multivariate linear regression. Furthermore, optimizing the step size of the integer-based approach is a simple linear regression problem (i.e., $\mathbf{w}^q = s \cdot \text{round}(\frac{\mathbf{w}}{s})$) while refining the step size of BiQ approach is a multiple linear regression problem (i.e., $\mathbf{w}^q = \alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2$). Thus, Mr.BiQ solves multivariate multiple regression problems. Since the multiple regression model is extended from linear regression model to include more than one independent variable, the multiple regression model is more accurate than the simple regression model, which explains the superior performance of Mr.BiQ over the conventional integer-based approaches.

C. Mr.BiQ with Uniform Quantization

We implement Mr.BiQ with uniform in an asymmetric way as BRECQ [4] and test it on Transformer models for vision tasks (see Table A4). By maintaining α_i to be $2 \times \alpha_{i+1}$, Mr.BiQ-Uniform can quantize models uniformly. For a fair comparison, we optimize the step size using straight through estimator (STE) [1] in BRECQ as well as bit-code (labeled "BRECQ-B" in Table A4). The results, however, do not show any significant performance gain; they rather show performance degradation in some cases. AdaRound [6] has also reported the difficulty of such a joint optimization. By reformulating the weights, Mr.BiQ-uniform can

optimize learnable parameters (α_i's and $\mathbf{b}_i$'s) as does in floating-point number without non-differentiable functions (e.g., round, floor, and sign). Which may lead to better results than the existing approach.

References

[1] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 1

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, volume 1, pages 4171–4186, 2019. 1

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015. 1

[4] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: Pushing the limit of post-training quantization by block reconstruction. In *International Conference on Learning Representations (ICLR)*, 2021. 1

[5] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *International Conference on Learning Representations (ICLR)*, 2017. 1

[6] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning (ICML)*, pages 7197–7206, 2020. 1

[7] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2383–2392, 2016. 1

[8] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1

[9] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018. 1

[10] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. 1

[11] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. MnasNet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2820–2828, 2019. 1

[12] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations (ICLR)*, 2018. 1