PQ-SAM: Post-training Quantization for Segment Anything Model

We thank all reviewers for your valuable time and for giving a number of positive comments on our paper. The main concerns are addressed as follows.

R1: The usefulness of your method for QAT. We would like to clarify that our method is a learnable PTQ, which indeed means that we do not involve any learning process for the model's parameters. Instead, our method involves learning activation's scaling and shifting sizes which are adaptively allocated by the OHC scheme based on the static activation distribution. Therefore, our method CANNOT be combined with QAT methods with dynamic activation distributions during fine-tuning. On the other hand, it is notable that QAT typically requires the whole labelled training dataset and substantial computational resources, which may not be affordable for large models. In contrast, our method is specifically designed for SAM and can be accomplished with just 100 calibration samples on two V100 GPUs. We will further discuss this issue in the updated paper.

R3: More ablation studies of heuristic optimization. Our method consists of three steps in order: ① tensor-wise truncation of outliers, ② channel-wise outlier grouping, and ③ optimized grouped shifting and scaling sizes. Steps ② and ③ have strict dependencies, where the order is crucial. We provide the ablation results of the exchanged order ablation for step ① and steps ②③ in the Table below.

Order	λ	6	10	14	α	0.20	0.25	0.30
① + ②③	IoU	0.6901	**0.7080**	0.6858	IoU	0.6884	**0.7080**	0.6841
②③+①	IoU	0.6350	**0.6810**	0.6692	IoU	0.6610	**0.6810**	0.6574

R3: The 0.63 tensor baseline may not be proper.

Kindly note that: 1) All our experiments were conducted under per-tensor quantization settings, which is more memory-efficient and computationally faster than per-channel quantization. 2) Although our method involves channel-wise activation scaling, the scaling sizes are reparameterized into weights during the per-tensor activation quantization. 3) As suggested, we add a comparison with a naive per-channel quantization method in the Table below, which shows the general superiority of our method.

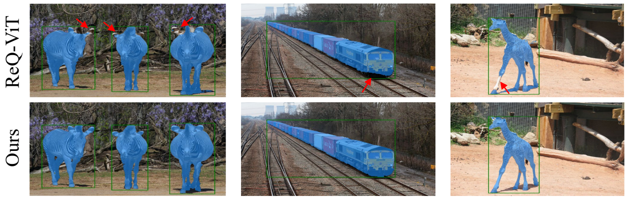
Δ_W, Δ_X	$\{v_n\}_{1 \le n \le N}$	$\{s_n\}_{1 \le n \le N}$	Per-Channel	IoU	Dice
✓	✗	✗	✗	0.6342	0.7334
✓	✗	✗	✓	0.6900	0.7750
✓	✓	✓	✗	**0.7080**	**0.7969**

R3,R5: Qauntization time Analysis.

Method	RepQ-Vit	LSQ+ [1]	Qdrop [2]	Ours
4 Bit (Iou / Time)	0.4738 / 0.1 h	0.4712/ 5h	0.5100 / 5h	**0.7080** / 5h

We provide a comparison of the performance and time of different methods in quantizing SAM-B in the Table above. While our method requires a longer quantization time due to learnable quantization parameters, it achieves a significant performance improvement compared to RepQ-Vit. Additionally, we compare our method with representative learning-based PTQ methods, such as LSQ+ and Qdrop. At a similar quantization time, our method still demonstrates clear performance advantages. This makes the quantization time acceptable in the field of large model compression, given the significant performance gains achieved.

R5,R8: More qualitative results. Given the length of the main paper, we have provided more qualitative results from SAM's point-prompt mode and automatic mode in the supplementary material. We further give more qualitative results from SAM's bounding-box prompt mode below.



R8: Discussion on the limitations of the method. The key limitation is its specific design for SAM models, which may require adjustments for other segmentation-based large models. We will emphasize the need for potential adaptations and modifications in the updated paper.

R8: Comparison with PTQ4SAM. Kindly note that PTQ4SAM was released on ArXiv on May 6, 2024, which was TWO months later than the submission of ECCV. Still, as suggested, we compare our PQ-SAM with PTQ4SAM in the following aspects: **(1)** Quantization Settings: Our PQ-SAM uses per-tensor quantization for weights and activations at 6-bit, 4-bit, and even 2-bit precision. In contrast, PTQ4SAM employs per-channel quantization for weights and post-softmax activations at 6-bit and 4-bit. **(2)** Methods: PTQ4SAM reduces quantization errors in key linear and softmax layers using a bimodal integration strategy. Our PQ-SAM mitigates activation outliers tensor-wise with GADT and OHC techniques. **(3)** Evaluation: Our PQ-SAM is evaluated in all SAM modes under zero-shot settings, while PTQ4SAM provides limited dataset-specific results, with unknown zero-shot capabilities. **(4)** Additionally, we give a quantitative comparison of the 4-bit quantized SAM-B model on the COCO dataset, utilizing the DINO detection head. Our PQ-SAM demonstrates significant superiority over PTQ4SAM. **(5)** We will add discussion of PTQ4SAM in the updated paper.

Method	FP	PTQ4SAM	PQ-SAM (Ours)
mAP	44.5	14.4	31.2

R9: Descriptions sometimes unclear. To demonstrate the effect of channel-wise grouping, we have provided a visualized analysis in Section 4 of the supplementary material. We will provide more descriptions in the updated paper.

[1] Lsq+: Improving low-bit quantization through learnable offsets and better initialization. In: CVPR workshop (2020)

[2] Multi-Scale 3D Gaussian Splatting for Anti-Aliased Rendering, Arxiv, 2023.