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

Xiaoyu Liu[1], Xin Ding[1], Lei Yu[2], Yuanyuan Xi[2], Wei Li[2], Zhijun Tu[2], Jie Hu[2], Hanting Chen[2], Baoqun Yin[1], and Zhiwei Xiong[1(✉)]

[1] University of Science and Technology of China
{liuxyu,xinding64}@mail.ustc.edu.cn, zwxiong@ustc.edu.cn
[2] Huawei Noah's Ark Lab
{wei.lee,zhijun.tu}@huawei.com

Abstract. Segment anything model (SAM) is a promising prompt-guided vision foundation model to segment objects of interest. However, the extensive computational requirements of SAM have limited its applicability in resource-constraint edge devices. Post-training quantization (PTQ) is an effective potential for fast-deploying SAM. Nevertheless, SAM's billion-scale pretraining creates a highly asymmetric activation distribution with detrimental outliers in excessive channels, resulting in significant performance degradation of the low-bit PTQ. In this paper, we propose PQ-SAM, the first PTQ method customized for SAM. To achieve a quantization-friendly tensor-wise distribution, PQ-SAM incorporates a novel grouped activation distribution transformation (GADT) based on a two-stage outlier hierarchical clustering (OHC) scheme to scale and shift each channel. Firstly, OHC identifies and truncates extreme outliers to reduce the scale variance of different channels. Secondly, OHC iteratively allocates learnable shifting and scaling sizes to each group of channels with similar distributions, reducing the number of learnable parameters and easing the optimization difficulty. These shifting and scaling sizes are used to adjust activation channels, and jointly optimized with quantization step sizes for optimal results. Extensive experiments demonstrate that PQ-SAM outperforms existing PTQ methods on nine zero-shot datasets, and pushes the 4-bit PTQ of SAM to a usable level.

Keywords: Segment Anything Model· Post-training Quantization

1 Introduction

The advent of large language models [2, 7, 52] has ushered in a new paradigm in natural language processing (NLP). These models exhibit exceptional zero-shot and few-shot generalization, learning broad competencies from sizable pretraining [14, 17, 33]. Inspired by their success, researchers have begun developing similarly large-scale foundation models for computer vision. A pioneering example

X. Liu and X. Ding—Equal contribution. This research was conducted while Xiaoyu Liu and Xin Ding were interns at Huawei Noah's Ark Lab.

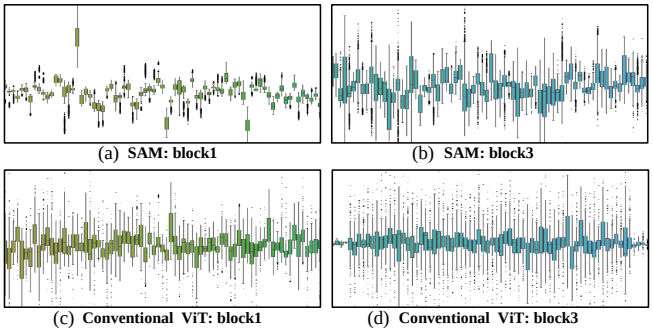

Fig. 1: Comparion of activation distributions between image encoder of SAM (a, b) and the conventional ViT (c, d) in two different transformer blocks. Both of these two models are based on the ViT-base version. Each distribution is visualized with the horizontal axis representing different channels and the vertical axis representing the intensity values within that channel.

is the recently proposed segment anything model (SAM) [19], representing the prompt-guided visual foundation model with impressive zero-shot capabilities. SAM focuses on the crucial task of segmenting objects of interest from an image, which underpins many vision applications [4, 11, 42, 45].

While SAM has demonstrated impressive advancements in segmentation abilities and flexibility, its practical deployment remains challenging. Specifically, SAM's reliance on image encoder of vision transformer (ViT) architecture induces extensive computational overhead that restricts applicability on edge devices with tight latency, power, and memory budgets. This problem necessitates the employment of model compression techniques such as network pruning [13, 49], knowledge distillation [15,26,27,47,48,53], model quantization [5,16,38] and compact architecture design [40, 43] *etc*. Among these techniques, post-training quantization (PTQ) only requires a few unlabeled calibration images without updating model parameters, which enables fast deployment on various devices within a limited time. Given that training and fine-tuning SAM requires significant GPUs using the large-scale training dataset SA1B [19] with 11 million images, there is a growing demand for the PTQ technique.

Nevertheless, exiting PTQ methods [22, 23, 25, 50] for ViT are focusing on conventional ViT models for classification, which are different from the ViT in SAM. SAM requires accurate prediction for each pixel on multiple zero-shot datasets after billion-scale pretraining, which is much more sensitive to low-bit compression for image embeddings, and its floating-point activations have a highly asymmetric distribution with detrimental outliers. As illustrated in Fig. 1, different from the conventional ViT, our analysis reveals two properties of SAM's activation distribution that pose challenges for quantization: The presence of detrimental outliers implies that the activation values in SAM are not symmetrically distributed around a central tendency. Moreover, SAM's activations often exhibit a concentration of extreme outliers in specific channels, resulting in

a considerable number of channels that require adjustment. While the existing PTQ methods [36, 41], such as RepQ-Vit [23], propose to scale activations at the channel level, the practical challenge in SAM arises in two aspects. Firstly, extreme outliers introduce detrimental scale variance in specific activation channels. Secondly, adjusting an excessive number of channels simultaneously proves to be challenging. These distribution properties of SAM's activations pose new challenges for existing PTQ methods, especially for the ultra-low-bit (*e.g.*, 4-bit, 2-bit) quantization.

In this paper, we propose PQ-SAM, the first PTQ method specially designed for SAM, incorporating a grouped activation distribution transformation (GADT) to reshape SAM's activations for quantization-friendly tensor-wise distribution guided by grouped learnable shifting and scaling sizes. Our proposed GADT has a specially designed outlier hierarchical clustering (OHC) scheme to suppress extreme outliers and reduce the total number of learnable parameters in GADT, easing optimization difficulty and preventing overfitting with limited training cost. Specifically, OHC first identifies and truncates tensor-wise extreme outliers across the whole activation tensor by examining interquartile range statistics at the tensor level. This effectively reduces the detrimental scale variance of different channels due to extreme outliers. Secondly, OHC iteratively allocates each pair of learnable shifting and scaling sizes for each group of channels with similar distributions by calculating their quantile range in each iteration. Critically, these shifting and scaling sizes are jointly optimized in an end-to-end manner together with the quantization step sizes for holistic optimal results, under the supervision of the full-precision (FP) SAM model. The optimized shifting and scaling sizes can be reparameterized into corresponding weights so that the proposed PQ-SAM is friendly for practical deployment.

We conduct extensive experiments to demonstrate the superiority of our PQ-SAM over existing advanced PTQ methods, on nine representative zero-shot datasets across three widely used segmentation modes of SAM, *i.e.*, point prompt mode, automatic mode, and bounding box prompt mode. Our PQ-SAM preserves SAM's strong zero-shot performance and achieves usable 4-bit PTQ for SAM, which represents an important step towards unlocking the practical edge deployment of SAM.

The contributions of this paper are as follows:

- We propose PQ-SAM, the first post-training quantization method customized for the segment anything model with limited computational cost.
- We propose a novel grouped activation distribution transformation (GADT) to reshape SAM's activations with learnable shifting and scaling sizes jointly end-to-end optimized together with quantization step sizes.
- We design a two-stage outlier hierarchical clustering scheme (OHC) to reduce the learnable parameters and ease the optimization difficulty in GADT.
- Extensive experiments demonstrate that PQ-SAM significantly outperforms previous PTQ methods on nine zero-shot datasets.

2 Related Work

2.1 Model Compression for SAM

To address the extensive computational requirements of SAM and make SAM suitable for edge devices, recent works MobileSAM [51] and FastSAM [54], EfficientSAM [44] propose different knowledge distillation based technological routes. MobileSAM reduces SAM's heavy image encoder through knowledge distillation. They replace SAM's encoder with a lightweight CNN and distil the heavyweight encoder's knowledge into the CNN. FastSAM avoids SAM's Transformer architecture entirely by adopting a regular convolution neural network (CNN) detector with a segmentation branch and retraining the CNN using 1/50th of SAM's SA1B dataset. EfficientSAM adopts a two-stage training strategy including preraining and finetuning.

While MobileSAM, FastSAM, and EfficientSAM can compress SAM models, they still require extensive SA1B data and training resources. In contrast, our proposed PQ-SAM technique only requires one millionth of SA1B data for calibration. With just 100 unlabeled images, PQ-SAM achieves efficient quantization of SAM with minimal GPU costs. PQ-SAM provides a highly practical solution to fast deploying SAM using commodity edge resources. Moreover, our PTQ method can be integrated into these methods for better deployment.

2.2 Model Quantization

Network quantization is an effective compression technique that transforms weight and activation values from floating points to low-bitwidth integers, substantially reducing memory, data movement, and energy costs. Existing methods fall into two categories: Quantization-Aware Training (QAT) [6,16,21] and Post-Training Quantization [10, 16]. However, QAT relies heavily on retraining with the full original dataset, which is infeasible for large vision models like SAM trained on massive data. While PTQ methods using limited calibration data are more practical, current techniques target classification and are not designed for SAM's unique segmentation task. Specifically, SAM exhibits a highly asymmetric distribution with detrimental outliers, which arises from its billion-scale pretraining. Prior PTQ methods for vision transformers [22, 25, 28, 50] do not address these characteristics of the large vision foundation model. To fill this gap, we develop the first customized PTQ approach, PQ-SAM, specifically tailored for quantizing the immense SAM model by accounting for its unusual activation properties.

3 Preliminary

To provide context before detailing our proposed method, we briefly introduce uniform quantization. We formulate k-bit quantization and dequantization of tensor x as

$$\begin{aligned} Quant: x_q &= \text{clip}\left(\text{round}\left(\frac{x}{\Delta}\right) + z, 0, 2^k - 1\right), \\ Dequant: x' &= \Delta\left(x_q - z\right) \approx x, \end{aligned} \tag{1}$$

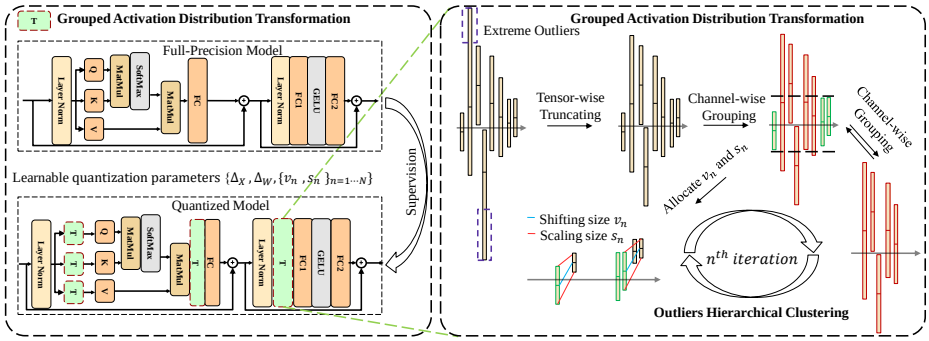

Fig. 2: The workflow of our proposed PQ-SAM. PQ-SAM introduces grouped activation distribution transformation (GADT) operations into each transformer block of SAM to enable model quantization.

where Round$(\cdot)$ projects a value to the k-bit nearest integer around the input, and $\text{clip}(x, l, u) = \min(\max(x, l), u)$. x_q represents the quantized value, and x' are de-quantized value which approximately equals to x. Note that $\Delta \in \mathbb{R}^+$ represents the quantization step size, and $z \in \mathbb{Z}$ denotes the zero-point which is a per-tensor offset. The values of Δ and z are determined based on the lower and upper bounds of x as follows:

$$\Delta = \frac{\max(x) - \min(x)}{2^k - 1}, \quad z = \text{round}\left(-\frac{\min(x)}{\Delta}\right). \tag{2}$$

Post-training quantization typically focuses on determining the quantization step sizes Δ_W and Δ_X for weights and activations per layer.

4 Grouped Activation Distribution Transformation

Overview. As shown in Fig. 2, PQ-SAM inserts learnable grouped activation distribution transformation (GADT) operations into each transformer block of SAM to reshape the distribution of activations. The shifting and scaling sizes in GADT are jointly optimized with quantization step sizes in an end-to-end manner under full-precision SAM supervision. This achieves holistically optimized quantization parameters tailored for SAM. Importantly, the learnable ADT is readily absorbed into the preceding linear layer, thus introducing no additional computations or parameters after quantization.

4.1 Scaling and Shifting Sizes

As shown in Fig. 1, the highly asymmetric activation distribution of SAM is problematic for the per-tensor quantization setting, making it challenging to properly determine the quantization parameters Δ_W and Δ_X. Moreover, the outliers concentrated in specific channels negatively bias Δ_W and Δ_X due to their disproportionate magnitude compared to the majority of activations. To

Algorithm 1: outlier hierarchical clustering

Input: FP SAM with activation $X \in \mathbb{R}^{T \times C_1}$
Output: $\{v_n, s_n\}_{1 \le n \le N}$

{ **Tensor-wise Outlier Truancting**.}
Count X_{Q1}, X_{Q3} and IQR of X
$X^t \leftarrow X$ in Eq. 5.
{ **Channel-wise Outlier Grouping**.}
Initialize the iteration number $n = 0$,
Allocated channel Group $C_a = \emptyset$,
Remaining channel Group $C_r = \{1, 2, ..., C_1\}$;
while C_r *is not* $\emptyset$ **do**
  foreach c *in* C_r **do**
    if $X_{1-\alpha}^{t,n-1} \le X^t(c) \le X_{\alpha}^{t,n-1}$ **then**
      Allocate $\{v_n, s_n\} \rightarrow X(c)$ in Eq. 6;
      C_a.push(c);
      C_r.pop(c);
    end
  end
  $n = n + 1$;
end

address this, we propose to transform SAM's activations towards a quantization-friendly distribution at the tensor level.

Straightforwardly, we introduce learnable channel-wise scaling size S and shifting size V to scale and shift the activation tensor on a channel-by-channel basis. For a linear layer, suppose that the input of an activation tensor $X \in \mathbb{R}^{T \times C_1}$ has T tokens and C_1 channels, and the weight matrix $W \in \mathbb{R}^{C_1 \times C_2}$ has a bias $B \in \mathbb{R}^{1 \times C_2}$. We utilize scaling size $S \in \mathbb{R}^{1 \times C_1}$ and shifting size $V \in \mathbb{R}^{1 \times C_1}$ to reshape its distribution, which is formulated as

$$\widetilde{X} = (X - V) \oslash S, \tag{3}$$

where $\widetilde{X}$ represents the reshaped activation tensor and $\oslash$ represents the division operation. This process is mathematically equivalent to the original linear layer by updating the weight matrix:

$$XW + B = \widetilde{X}\left(W^\top \odot S^\top\right) + \left(VW^\top + B\right), \tag{4}$$

where $\odot$ represents the multiplication operation, and the updated linear layer has a weight matrix $\widetilde{W} = W^\top \odot S^\top$ and a bias $\widetilde{B} = \left(VW^\top + B\right)$.

Given the practical challenge of optimizing SAM's large number of activation channels, it is difficult to simultaneously optimize the shifting and scaling sizes at the channel level. To address this, we propose Grouped Activation Distribution Transformation (GADT). This technique utilizes our outlier hierarchical clustering scheme to group channels with similar distributions together. By sharing a set of learnable parameters within each group, we effectively reduce the optimization difficulty.

4.2 Outlier Hierarchical Clustering

The above-mentioned shifting and scaling sizes are one-on-one corresponding to each activation channel, which leads to excessive learned parameters being optimized and increases optimization difficulty within limited training cost. To reduce learning difficulty and prevent overfitting of the learnable shifting and scaling sizes, we propose an outlier hierarchical clustering (OHC) scheme to iteratively allocate a pair of shifting and scaling sizes to a group of channels. OHC consists of a tensor-wise outlier truncating stage and a channel-wise outlier grouping stage.

Tensor-wise Outlier Truncating. Due to the influence of extreme outliers in the activation values, the grouping of channels with similar distributions can be affected. Additionally, these extreme anomalies can impact the effective usage of quantization intervals. Therefore, as a preliminary step, we perform outlier truncation at the tensor level by examining the outlier situation of numerical values across the entire tensor layer.

Motivated by the statistic theory [34,39,55], we design an effective preprocessing scheme. For the activation tensor X, we count the upper and lower quartile fractions Q_3 and Q_1, and the interquartile range IQR of its numerical distribution. We then introduce adjustable statistical thresholds, and the truncated activation X^t is formulated as

$$\begin{aligned} X^t = \{X | X_{Q_1} - \lambda IQR < x < X_{Q_3} + \lambda IQR, x \in X\}, \\ IQR = X_{Q_3} - X_{Q_1}, \end{aligned} \tag{5}$$

where λ is a hyper-parameter, and X_{Q_3} and X_{Q_1} are the upper and lower quartile values at Q_3 and Q_1, respectively.

Channel-wise Outlier Grouping. For a truncated activation tensor X_t consisting of C channels, we allocate N pairs of shifting and scaling sizes $\{v_n, s_n\}_{1 \leq n \leq N}$ to the channels, and $N \ll C$. As depicted in Fig. 2 (b), we perform this allocation iteratively, assigning a pair of shifting and scaling sizes v_n, s_n to a group of channels that meet the pre-defined range between the upper quantile α and lower quantile $1 - \alpha$ of the non-allocating channels. In the n^{th} iteration, we formulate the procedure of $\{v_n, s_n\}$ for activations of the c^{th} channel $X(c)$ as

$$\widetilde{X}(c) = (X(c) - v_n) \oslash s_n, \text{ if } X_{1-\alpha}^{t,n-1} \leq X^t(c) \leq X_{\alpha}^{t,n-1}, \tag{6}$$

where $\widetilde{X}(c)$ represents the the c^{th} channel of reshaped activation $\widetilde{X}$. Additionally, $X_{1-\alpha}^{t,n-1}$ and $X_{\alpha}^{t,n-1}$ respectively denote the lower and upper quantile values, of the activations from these channels that are not allocated in previous iterations.

Overall, the procedure of the proposed outlier hierarchical clustering is summarized in Algorithm 1. This iterative allocation process allows for the dynamic calculation of percentile values for the unallocated channels. By using a fixed value of α, we ensure that more similar channels are grouped together at the beginning of the allocation process. As the allocation process advances, the groups gradually become more refined, resulting in finer granularity. This adaptive approach ultimately yields highly effective grouping outcomes, which prove advantageous for subsequent learning and optimization. By conducting the joint

optimization of GADT and quantization step sizes, PQ-SAM quantizes SAM with limited overheads on unlabeled samples.

4.3 Joint Optimization with Quantization Step Sizes

The shifting and scaling sizes in GADT are learnable and optimized jointly with Δ_W and Δ_X in our PQ-SAM. This integration of learnable shifting and scaling sizes allows for a more effective adjustment of the activation distribution. We use FP model supervision to learn the quantization step sizes Δ_W and Δ_X in an end-to-end manner, and adjust offsets via the above-mentioned GADT, which not only achieves channel-wise offset adjustment but also preserves the hardware-friendly per-tensor quantization setting.

Given the floating-point weight tensor W and activation tensor X in one layer of SAM, we follow the Eq. 1 and quantize them to k-bit integer values W_q and X_q with the corresponding weight and activation quantization step sizes Δ_W and Δ_X, respectively. Our optimization framework goes beyond previous approaches [23,36,41] that focus on determining Δ_W and Δ_X for each layer individually. Instead, our framework enables a holistic, FP-supervised quantization process that is customized for the entire model.

We jointly optimize the shifting and scaling sizes $\{v_n, s_n\}_{1\leq n\leq N}$, and quantization step sizes Δ_W and Δ_X for holistic optimal results. By leveraging the output $O(W, K)$ of the FP model as a guide, we can tailor the quantization steps specifically for the pre-trained weights and activations by minimizing the Mean Squared Error (MSE) loss between $O(W, X)$ and the output $O(\widetilde{W}_q, \widetilde{X}_q)$ of quantized model:

$$\underset{\Delta_W, \Delta_X, \{v_n, s_n\}_{1\leq n\leq N}}{\arg\min} ||O(\widetilde{W}_q, \widetilde{X}_q) - O(W, X)||_2, \tag{7}$$

where $\widetilde{W}_q$ represent the corresponding quantized weight of the updated weight by reparameterization as demonstrated in Eq. 4, and $\widetilde{X}_q$ represents the quantized activation which has been reshaped.

5 Experiment

5.1 Datasets and Metrics

We conduct evaluations on nine representative zero-shot segmentation datasets, focusing on three widely used segmentation modes of SAM, *i.e.*, the point prompt mode, the automatic mode, and the bounding box prompt mode. These datasets have been officially verified by SAM as mentioned in the work [19]. However, since there is no publicly available test code provided by SAM, we independently evaluate these datasets based on our own resources.

Point Prompt Mode. The point prompt specifies what to segment in the image, with the goal of determining the mask containing that point. To fully validate the zero-shot performance of quantized SAM, we subsample six datasets

Table 1: Quantitative comparison of different PTQ methods for the point prompt mode of SAM.

	Method	Bit	BBBC038V1		NDD20		Cityscape		DOORS		iShape		NDISPark	
			IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice
	FP	32	0.7795	0.8500	0.8468	0.9100	0.5945	0.7061	0.8496	0.9126	0.3722	0.5108	0.8192	0.8938
ViT-H	MinMax [16]	6	0.3485	0.4599	0.5394	0.6605	03328	0.4374	0.3678	0.5082	0.2590	0.3863	0.6095	0.7091
	MSE [5]	6	0.6598	0.7393	0.5511	0.6375	0.4423	0.5518	0.6115	0.7076	0.3409	0.4818	0.7136	0.8046
	Percentile [20]	6	0.7219	0.7290	0.7668	0.8494	0.5053	0.6155	0.7871	0.8617	**0.3679**	**0.5076**	0.7850	0.8672
	FQ-ViT [25]	6	0.4444	0.6008	0.2513	0.3805	0.2836	0.4186	0.2867	0.4298	0.1088	0.1926	0.4137	0.5715
	RepQ-ViT [23]	6	0.7174	0.7938	0.8223	**0.8939**	0.5598	0.6785	0.8383	0.9048	0.3444	0.4889	0.8010	0.8838
	Ours	6	**0.7646**	**0.8376**	**0.8225**	0.8933	**0.5697**	**0.6861**	**0.8451**	**0.9092**	0.3380	0.4776	**0.8157**	**0.8919**
	MinMax [16]	4	0.3309	0.4638	0.2718	0.4050	0.1920	0.2966	0.4634	0.6144	0.1133	0.2007	0.2515	0.3735
	MSE [5]	4	0.3761	0.5141	0.3646	0.5187	0.2265	0.3394	0.4265	.05835	0.1445	0.2473	0.3933	0.5408
	Percentile [20]	4	0.3735	0.5200	0.3447	0.4811	0.2619	0.3891	0.3888	0.5273	0.1253	0.2169	0.4153	0.5657
	FQ-ViT [25]	4	0.2213	0.3484	0.0972	0.1636	0.1453	0.2387	0.1168	0.1983	0.0359	0.0678	0.2103	0.3341
	RepQ-ViT [23]	4	0.7013	0.7809	0.7685	0.8550	0.4590	0.5740	0.7590	0.8493	0.2736	0.4065	0.7399	0.8325
	Ours	4	**0.7081**	**0.7973**	**0.7818**	**0.8670**	**0.5056**	**0.6284**	**0.7975**	**0.8766**	**0.2921**	**0.4294**	**0.7647**	**0.8541**
	FP	32	0.7761	0.8476	0.8468	0.9075	0.5939	0.7068	0.8514	0.9138	0.3479	0.4856	0.8127	0.8893
ViT-L	MinMax [16]	6	0.5129	0.6285	0.5262	0.6387	0.3681	0.4838	0.4196	0.5612	0.2821	0.4144	0.6490	0.7532
	MSE [5]	6	0.7010	0.7764	0.6521	0.7408	0.4757	0.5888	0.6857	0.7808	0.3544	0.4964	0.7583	0.8489
	Percentile [20]	6	0.7368	0.8066	0.7511	0.8320	0.5230	0.6364	0.7805	0.8574	0.3775	0.5195	0.7940	0.8759
	FQ-ViT [25]	6	0.4213	0.5660	0.3596	0.5143	0.2840	0.4206	0.3840	0.5453	0.1205	0.2113	0.4098	0.5675
	RepQ-ViT [23]	6	0.6952	0.7715	0.8004	0.8767	**0.5580**	0.6727	0.8270	0.8940	0.3105	0.4486	0.7868	0.8712
	Ours	6	**0.7425**	**0.8154**	**0.8180**	**0.8902**	0.5566	**0.6735**	**0.8370**	**0.9036**	**0.3199**	**0.4593**	**0.7990**	**0.8786**
	MinMax [16]	4	0.1256	0.2053	0.2764	0.4161	0.1285	0.2027	0.4151	0.5775	0.0948	0.1707	0.1818	0.2868
	MSE [5]	4	0.2836	0.4042	0.3304	0.4747	0.1841	0.2766	0.3341	0.4901	0.1554	0.2643	0.3641	0.4994
	Percentile [20]	4	0.4136	0.5637	0.3074	0.4531	0.2720	0.4053	0.3701	0.5259	0.1355	0.2339	0.4084	0.5631
	FQ-ViT [25]	4	0.2791	0.4238	0.1669	0.2747	0.1794	0.2903	0.1813	0.2979	0.0746	0.1376	0.2896	0.4406
	RepQ-ViT [23]	4	0.4129	0.4983	0.7235	0.8250	0.4422	0.5622	0.7552	0.8483	**0.2791**	**0.4103**	0.7100	0.8118
	Ours	4	**0.6855**	**0.7727**	**0.7393**	**0.8339**	**0.4878**	**0.6114**	**0.7890**	**0.8712**	0.2520	0.3833	**0.7379**	**0.8332**
	FP	32	0.7666	0.8417	0.8322	0.9000	0.5692	0.6884	0.8294	0.8987	0.3536	0.4987	0.7933	0.8760
ViT-B	MinMax [16]	6	0.3829	0.4777	0.5727	0.6976	0.3356	0.4445	0.3495	0.4797	0.2685	0.4045	0.5904	0.7021
	MSE [5]	6	0.6325	0.7255	0.5767	0.6921	0.3976	0.5163	0.5030	0.6346	0.2943	0.4346	0.6294	0.7398
	Percentile [20]	6	0.7269	0.7997	0.7364	0.8260	0.5199	0.6406	0.7952	0.8747	0.3605	0.5068	0.7676	0.8564
	FQ-ViT [25]	6	0.6154	0.7152	0.4783	0.6010	0.3826	0.5217	0.3538	0.4859	0.2083	0.3310	0.5846	0.7196
	RepQ-ViT [23]	6	0.6900	0.7675	**0.8070**	0.8730	**0.5449**	**0.6644**	0.8053	0.8801	0.3439	0.4856	0.7836	0.8696
	Ours	6	**0.7583**	**0.8331**	0.8008	**0.8782**	0.5391	0.6593	**0.8059**	**0.8886**	**0.3441**	**0.4886**	**0.7845**	**0.8705**
	MinMax [16]	4	0.0556	0.0986	0.1533	0.2413	0.0897	0.1374	0.1397	0.2252	0.0447	0.0842	0.0589	0.0964
	MSE [5]	4	0.3359	0.4612	0.3756	0.5315	0.2278	0.3343	0.4635	0.6263	0.1330	0.2306	0.3637	0.5018
	Percentile [20]	4	0.3927	0.5381	0.4054	0.5583	0.2728	0.4034	0.4219	0.5761	0.1356	0.2345	0.4149	0.5564
	FQ-ViT [25]	4	0.3476	0.5010	0.1669	0.2747	0.2101	0.3292	0.2284	0.3591	0.1134	0.1995	0.3119	0.4647
	RepQ-ViT [23]	4	0.4738	0.5659	0.6479	0.7608	0.4338	0.5653	0.5878	0.7118	0.3064	0.4466	0.6253	0.7448
	Ours	4	**0.7080**	**0.7969**	**0.6768**	**0.7837**	**0.4714**	**0.5986**	**0.7375**	**0.8361**	**0.3106**	**0.4528**	**0.6800**	**0.7983**

including BBBC038V1 [3], NDD20 [37], Cityscape [9], DOORS [32], iShape [46], and NDISPark [8]. These datasets contain 15,213 images and 29,978 masks and cover a broad range of domains, which is used for the very comprehensive evaluation experiment. Following the original SAM's experiments, we determine the single-point prompt by selecting the "center" of the ground truth. This center is identified as the point with the highest value in the mask's interior distance transform. We calculate the mean IoU and Dice [35] for all masks of each dataset as quantitative metrics, which are common metrics to quantify the overlap between predicted and ground truth masks.

Automatic Mode. SAM's automatic mask generation pipeline can automatically extract a series of masks from an input image with a regular grid of foreground points. We evaluate this automatic mode on the zero-shot edge detection task using the BSDS500 [30] dataset. These automatically generated masks are processed into object edges by the post-processing [1]. We conduct evaluations on the test subset, consisting of 200 images, and assess the performance of all methods using four standard metrics [2] for edge detection [1]: optimal dataset

[1] Our experiments are based on the descriptions in SAM's paper since the official test code is not provided to the public.

[2] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

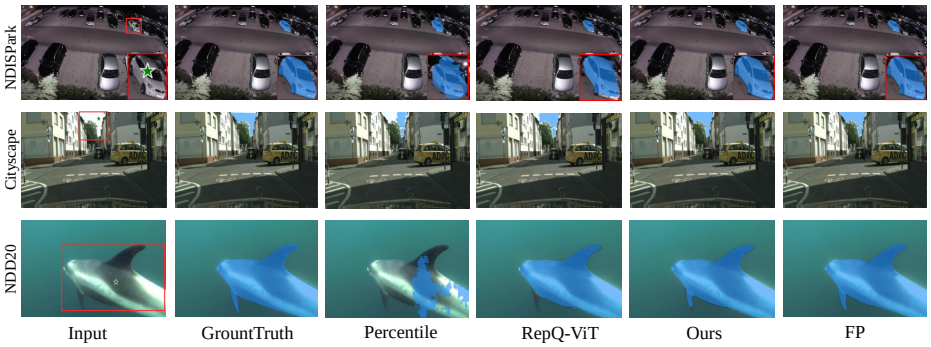

Fig. 3: Visual comparison of different PTQ methods for SAM's point prompt mode. The red block and green star represent the object to be segmented and the position of the prompt point, respectively.

scale (ODS), optimal image scale (OIS), average precision (AP), and recall at 50% precision (R50).

Bounding Box Prompt Mode. Similar to the point prompt mode, the bounding box prompt mode of SAM focuses on segmenting the objects circled by the bounding box. We follow the original setting of SAM [19] and conduct experiments on the COCO [24] and LVIS [12] datasets. The results are evaluated by average precision (AP).

5.2 Implementation Details

Training Details. We utilize the PyTorch [31] and conduct all of our experiments on NVIDIA Tesla V100 GPUs. To calibrate our model, we randomly select 100 samples from the SA1B dataset as calibration data and perform the calibration process over 100 epochs, using a batch size of 1. The learnable quantization parameters Δ_W, Δ_X, and $\{v_n, s_n\}_{1 \ll n \ll N}$, are optimized using the Adam optimizer [18] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, with a learning rate of $1e-4$. The learning rate scheduler is CosineAnnealingLR [29]. The hyperparameters λ and α are set as 10 and 0.25, respectively.

Experimental Settings. In our evaluation experiments, we focus on challenging low-bit-width quantization settings, specifically W4A4 (4-bit quantization for weight and activation) and W6A6 (6-bit quantization), for the point prompt mode of SAM. It is important to note that the automatic mode of SAM, which is used for the edge detection task, is less sensitive to network prediction errors due to the incorporation of various post-processing algorithms. Therefore, we specifically perform 4-bit quantization experiments on the BSDS500 dataset for the automatic mode of SAM. Additionally, to validate the superiority of our PQ-SAM approach as discussed in Section 5.3, we conduct experiments in an ultra-low 2-bit setting specifically for the ViT-B version of SAM. This ultra-low-bit setting allowed us to assess the performance and effectiveness of PQ-SAM in scenarios with extremely limited precision.

Baseline Methods. In our comparative analysis, we include two categories of PTQ methods as baselines. The first category consists of classical post-training

Table 2: Quantitative comparison for the automatic mode of SAM.

	Method	Bit	ODS	OIS	AP	R50
ViT-H	FP	32	0.754	0.770	0.729	0.865
	MinMax [16]	4	0.430	0.440	0.304	-
	MSE [5]	4	0.454	0.468	0.388	0.049
	Percentile [20]	4	0.443	0.457	0.339	-
	FQ-ViT [25]	4	0.412	0.416	0.230	-
	RepQ-ViT [23]	4	0.706	0.720	0.647	0.801
	Ours	4	**0.736**	**0.745**	**0.707**	**0.859**
ViT-L	FP	32	0.754	0.773	0.726	0.866
	MinMax [16]	4	0.423	0.424	0.247	-
	MSE [5]	4	0.442	0.456	0.332	-
	Percentile [20]	4	0.444	0.455	0.320	-
	RepQ-ViT [23]	4	0.709	0.722	0.645	0.769
	FQ-ViT [25]	4	0.424	0.429	0.255	-
	Ours	4	**0.726**	**0.735**	**0.669**	**0.843**
ViT-B	FP	32	0.745	0.755	0.690	0.871
	MinMax [16]	4	0.356	0.357	0.145	-
	MSE [5]	4	0.425	0.480	0.475	0.374
	Percentile [20]	4	0.449	0.467	0.400	0.024
	FQ-ViT [25]	4	-	-	-	-
	RepQ-ViT [23]	4	0.701	**0.712**	0.632	0.806
	Ours	4	**0.706**	0.710	**0.642**	**0.861**

methods, namely MSE [5] and MinMax [16]. These methods are versatile and can be easily applied to different models. The second category comprises quantization methods specifically designed for Vision Transformer (ViT) models, including two state-of-the-art methods RepQ-ViT [23] and FQ-ViT [25]. These methods have gained popularity in post-training quantization and are directly relevant to our evaluation. Specifically, RepQ-ViT is designed for the post-LayerNorm activations with severe inter-channel variation but based on a hand-craft heuristic algorithm, which is not as flexible and effective as our learning-based method.

5.3 Experimental Results

Point Prompt Mode. In Fig. 3, we present the quantitative results of different post-quantization methods for the SAM's point prompt mode. Overall, our PQ-SAM consistently outperforms existing methods across most datasets, which proves that our method can better maintain the strong generalization capacity of SAM. For the 6-bit quantization, PQ-SAM exhibits a performance gap to the full-precision model of less than 0.04 in terms of IoU and Dice metrics across most of the datasets. While PQ-SAM may exhibit slightly lower performance compared to strong baseline methods on a few datasets, it still demonstrates competitive results and preserves the generalization ability overall. For the 4-bit quantization, our method significantly surpasses the baseline methods across all datasets. For instance, PQ-SAM surpasses the second-best method by over 40% in terms of IoU and Dice metrics on the BBBC038V1 dataset for the SAM's ViT-L and ViT-B versions.

Automatic Mode. In Table 2, we present the quantitative results of different post-quantization methods for the SAM's automatic mode on the BSDS500 dataset. Our PQ-SAM surpasses the baseline methods across three SAM versions and archives a small performance gap to the FP model.

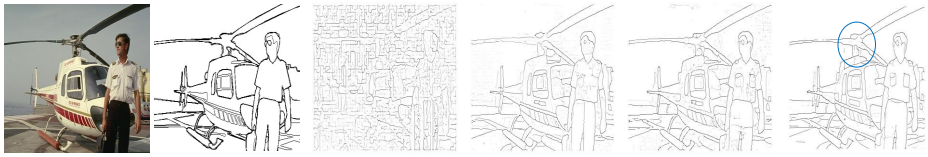

Fig. 4: Visual comparison of different PTQ methods for SAM's automatic mode. The blue circle indicates segmentation errors.

Table 3: Quantitative Results for the bounding box prompt mode of SAM.

	Method	Bit	COCO				LVIS			
			AP	AP^S	AP^M	AP^L	AP	AP^S	AP^M	AP^L
VIT-H	FP	32	46.4	30.7	51.0	61.7	44.4	32.1	57.2	65.2
	RepQ-Vit [23]	6	43.5	27.6	48.1	59.2	42.4	30.0	54.9	63.8
	Ours	6	**45.1**	**29.6**	**49.5**	**60.2**	**43.4**	**31.5**	**55.7**	**64.0**
	RepQ-Vit [23]	4	38.2	24.7	41.7	52.5	36.0	25.5	46.5	54.1
	Ours	4	**41.1**	**26.2**	**45.3**	**55.9**	**38.1**	**27.1**	**49.1**	**56.9**

Table 4: Quantitative comparison for the ultra-low 2bit quantization setting.

Method	NDD20		DOORs		NDISPark	
	IoU	Dice	IoU	Dice	IoU	Dice
RepQ-ViT [23]	0.2681	0.4146	0.3677	0.5301	0.2780	0.4224
Ours	**0.2941**	**0.4480**	**0.4099**	**0.5697**	**0.4213**	**0.5812**

Bounding Box Prompt Mode. In Table 3, we provide the quantitative results of different post-quantization methods for the SAM's bounding box prompt on the COCO and LVIS datasets. Our PQ-SAM outperforms the baseline methods and has a small performance gap compared to the FP model. In the 6-bit quantization setting, PQ-SAM exhibits a loss of accuracy within 3% compared to the FP model. Moreover, in the 4-bit quantization configuration, PQ-SAM shows a significant improvement of 6% compared to the RepQ-Vit method.

Ultra-low Bit Setting. As shown in Table 4, we validate the superiority of our PQ-SAM on the ultra-low 2-bit quantization setting, which significantly surpasses the state-of-the-art post-training quantization (PTQ) method RepQ-ViT. For instance, our method improves Dice by over 35% compared to RepQ-ViT, demonstrating the effectiveness of our approach for the ultra-low-bit quantization of SAM models.

Visualization. We provide the visual results of quantized SAM using different quantization methods for the W4A4 setting, including the point prompt mode in Fig. 3 and the auto mode in Fig. 4, respectively. Compared with existing PTQ methods, our proposed method could achieve better segmentation performance with more accurate segmentation masks and fewer segmentation errors. We further give more qualitative results in the supplementary material.

5.4 Ablation Studies

We perform ablation studies to evaluate the impacts of each component in the proposed PQ-SAM approach using SAM's ViT-B on the BBBC039V1 dataset.

Table 5: Ablation study on learnable quantization parameters. δ_W and δ_X represent the weight and activation quantization step sizes, respectively. $\{v_n, s_n\}_{1 \leq n \leq N}$ represents the allocated N pairs of shifting and scaling sizes.

Δ_W,Δ_X	$\{v_n\}_{1 \leq n \leq N}$	$\{s_n\}_{1 \leq n \leq N}$	IoU	Dice
✘	✘	✘	0.0556	0.0986
✘	✔	✔	0.1515	0.2339
✔	✘	✘	0.6342	0.7334
✔	✔	✘	0.6847	0.7721
✔	✔	✔	**0.7080**	**0.7969**

Table 6: Ablation study on activation distribution transformation (ADT) with the outlier hierarchical clustering (OHC). 'Truncating' represents the tensor-level outlier truncating operation. 'Even Groupsize' represents that we evenly allocate the same number of learnable shifting and scaling sizes as ours to each group of adjacent channels. 'Per-Channel' represents that we allocate each pair of learnable shifting and scaling sizes to each channel.

ADT	Truncating	Even Groupwise	Per-Channel	OHC (ours)	IoU	Dice
✘	✘	✘	✘	✘	0.6342	0.7334
✘	✔	✘	✘	✘	0.6533	0.7602
✔	✔	✘	✔	✘	0.6842	0.7720
✔	✔	✔	✘	✘	0.6817	0.7797
✔	✔	✘	✘	✔	**0.7080**	**0.7969**

Learnable Quantization Parameters. As presented in Table 5, we perform an ablation study on the learnable quantization parameters in our PQ-SAM framework. This involves considering the weight and activation quantization step sizes, denoted as δ_W and δ_X respectively, as well as allocating a set of N pairs of shifting and scaling sizes $\{v_n, s_n\}_{1 \leq n \leq N}$.

The results demonstrate that introducing learnable quantization step sizes, shifting sizes, or scaling sizes individually resulted in significantly improved performance compared to fixed quantization schemes. Moreover, it is observed that the learnable shifting and scaling sizes yielded even better quantization performance than the learnable quantization step sizes. By optimizing all of these quantization parameters during training, the model is able to effectively adapt to the underlying data distribution and achieve more accurate quantization.

Activation Distribution Transformation. In Table 6, we present the results of an ablation study to validate the effectiveness of the activation distribution transformation (ADT) in combination with the outlier hierarchical clustering (OHC) scheme. The ADT technique significantly improves the quantization performance, and the OHC scheme further enhances the optimization process and leads to additional improvements in quantization performance. Meanwhile, we validate that the tensor-level outlier truncating operation in ADT can significantly improve the quantization performance.

To validate the effectiveness of the outlier hierarchical clustering (OHC) scheme, we compare it with two alternative schemes: the "Even Groupsize" and "Per-Channel" schemes. In the "Even Groupsize" scheme, we allocate the same number of learnable shifting and scaling sizes to each group of adjacent channels. In the "Per-Channel" scheme, we allocate a unique pair of learnable shifting and scaling sizes to each individual channel, resulting in the same number of learn-

Table 7: Ablation study on hyper-parameters and amounts of calibration data.

(a) Hyper-parameters of PQ-SAM.

λ	6	10	14
IoU	0.6901	**0.7080**	0.6858
Dice	0.7894	**0.7969**	0.7871
α	0.20	0.25	0.30
IoU	0.6884	**0.7080**	0.6841
Dice	0.7812	**0.7969**	0.7693

(b) Amount of calibration data.

Number of data	100	200	500
IoU	0.7080	0.7091	0.7120
Dice	0.7969	0.7980	0.8014

able shifting and scaling size pairs as there are activation channels. The results of our ablation study demonstrate that the activation distribution transformation (ADT) with OHC in the PQ-SAM framework outperforms those with the "Even Groupsize" or "Per-Channel" schemes.

Hyperparameter λ and α. We conduct ablation studies with extreme parameter settings to showcase the impact of hyperparameters λ and α on performance, as shown in Table 7a. The first set of experiments focuses on evaluating the impact of the hyperparameter λ on tensor-level outlier truncation. Different values of λ are tested for various truncated ranges. The results demonstrate the effects of varying λ on the outlier truncation process and its subsequent influence on the overall performance of the PQ-SAM framework. The second set of experiments involves an ablation study on different upper quantiles α for the proposed channel-wise outlier grouping in the outlier hierarchical clustering step. By testing different values of α, different numbers of learnable shifting and scaling sizes are allocated to each group of channels.

Amount of Calibration Data. We conduct an ablation study on the amount of calibration data. The results in Table 7b show that 100 samples are sufficient for our method, and additional data does not yield significant improvements.

6 Conclusion

Our proposed PQ-SAM addresses the challenge of resource constraints in deploying the segment anything model on edge devices. By incorporating the grouped activation distribution transformation with outlier hierarchical clustering techniques, PQ-SAM offers a customized post-training quantization (PTQ) solution for SAM. The GADT reshapes the asymmetric activation distribution of SAM, while the OHC scheme effectively handles extreme outliers and optimizes the allocation of shifting and scaling sizes for each group of channels. Our extensive experiments demonstrate that PQ-SAM surpasses existing PTQ methods on nine zero-shot datasets across three modes of SAM, especially for the low-bit quantization settings.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China under Grants 62131003 and 62021001. We gratefully acknowledge the support of MindSpore, CANN (Compute Architecture for Neural Networks) and Ascend AI Processor used for this research.

References

1. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE transactions on pattern analysis and machine intelligence **33**(5), 898–916 (2010)
2. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. NeurIPS (2020)
3. Caicedo, J.C., Goodman, A., Karhohs, K.W., Cimini, B.A., Ackerman, J., Haghighi, M., Heng, C., Becker, T., Doan, M., McQuin, C., et al.: Nucleus segmentation across imaging experiments: the 2018 data science bowl. Nature methods **16**(12), 1247–1253 (2019)
4. Cheng, H.K., Oh, S.W., Price, B., Schwing, A., Lee, J.Y.: Tracking anything with decoupled video segmentation. In: ICCV (2023)
5. Choi, J., Chuang, P.I.J., Wang, Z., Venkataramani, S., Srinivasan, V., Gopalakrishnan, K.: Bridging the accuracy gap for 2-bit quantized neural networks (qnn). arXiv preprint arXiv:1807.06964 (2018)
6. Choi, J., Wang, Z., Venkataramani, S., Chuang, P.I.J., Srinivasan, V., Gopalakrishnan, K.: Pact: Parameterized clipping activation for quantized neural networks. arXiv preprint arXiv:1805.06085 (2018)
7. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311 (2022)
8. Ciampi, L., Santiago, C., Costeira, J.P., Gennaro, C., Amato, G.: Domain adaptation for traffic density estimation. In: VISAPP (2021)
9. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
10. Ding, X., Liu, X., Zhang, Y., Tu, Z., Li, W., Hu, J., Chen, H., Tang, Y., Xiong, Z., Yin, B., et al.: Cbq: Cross-block quantization for large language models. arXiv preprint arXiv:2312.07950 (2023)
11. Gong, S., Zhong, Y., Ma, W., Li, J., Wang, Z., Zhang, J., Heng, P.A., Dou, Q.: 3dsam-adapter: Holistic adaptation of sam from 2d to 3d for promptable medical image segmentation. arXiv preprint arXiv:2306.13465 (2023)
12. Gupta, A., Dollar, P., Girshick, R.: Lvis: A dataset for large vocabulary instance segmentation. In: CVPR (2019)
13. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149 (2015)
14. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: CVPR (2022)
15. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
16. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: CVPR (2018)
17. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: ICML (2021)

18. Kingma, D.P., Ba, J.L.: Adam: A method for stochastic optimization. In: ICLR (2015)
19. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: ICCV (2023)
20. Li, R., Wang, Y., Liang, F., Qin, H., Yan, J., Fan, R.: Fully quantized network for object detection. In: CVPR (2019)
21. Li, Z., Gu, Q.: I-vit: Integer-only quantization for efficient vision transformer inference. In: ICCV (2023)
22. Li, Z., Ma, L., Chen, M., Xiao, J., Gu, Q.: Patch similarity aware data-free quantization for vision transformers. In: ECCV (2022)
23. Li, Z., Xiao, J., Yang, L., Gu, Q.: Repq-vit: Scale reparameterization for post-training quantization of vision transformers. In: ICCV (2023)
24. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
25. Lin, Y., Zhang, T., Sun, P., Li, Z., Zhou, S.: Fq-vit: Post-training quantization for fully quantized vision transformer. arXiv preprint arXiv:2111.13824 (2021)
26. Liu, X., Cai, M., Chen, Y., Zhang, Y., Shi, T., Zhang, R., Chen, X., Xiong, Z.: Cross-dimension affinity distillation for 3d em neuron segmentation. In: CVPR (2024)
27. Liu, X., Hu, B., Huang, W., Zhang, Y., Xiong, Z.: Efficient biomedical instance segmentation via knowledge distillation. In: MICCAI (2022)
28. Liu, Z., Wang, Y., Han, K., Zhang, W., Ma, S., Gao, W.: Post-training quantization for vision transformer. In: NeurIPS (2021)
29. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
30. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV (2001)
31. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS (2019)
32. Pugliatti, M., Topputo, F.: Doors: Dataset for boulders segmentation. Zenodo **9**, 20 (2022)
33. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
34. Ross, S.M.: Introductory statistics. Academic Press (2017)
35. Shamir, R.R., Duchin, Y., Kim, J., Sapiro, G., Harel, N.: Continuous dice coefficient: a method for evaluating probabilistic segmentations. arXiv preprint arXiv:1906.11031 (2019)
36. Shao, W., Chen, M., Zhang, Z., Xu, P., Zhao, L., Li, Z., Zhang, K., Gao, P., Qiao, Y., Luo, P.: Omniquant: Omnidirectionally calibrated quantization for large language models. In: ICLR (2023)
37. Trotter, C., Atkinson, G., Sharpe, M., Richardson, K., McGough, A.S., Wright, N., Burville, B., Berggren, P.: Ndd20: A large-scale few-shot dolphin dataset for coarse and fine-grained categorisation. arXiv preprint arXiv:2005.13359 (2020)
38. Tu, Z., Hu, J., Chen, H., Wang, Y.: Toward accurate post-training quantization for image super resolution. In: CVPR (2023)
39. Upton, G., Cook, I.: Understanding statistics. Oxford University Press (1996)
40. Vasu, P.K.A., Gabriel, J., Zhu, J., Tuzel, O., Ranjan, A.: Fastvit: A fast hybrid vision transformer using structural reparameterization. In: ICCV (2023)

41. Wei, X., Zhang, Y., Li, Y., Zhang, X., Gong, R., Guo, J., Liu, X.: Outlier suppression+: Accurate quantization of large language models by equivalent and effective shifting and scaling. In: EMNLP (2023)
42. Xie, D., Wang, R., Ma, J., Chen, C., Lu, H., Yang, D., Shi, F., Lin, X.: Edit everything: A text-guided generative system for images editing. arXiv preprint arXiv:2304.14006 (2023)
43. Xie, Y., Liao, Y.: Efficient-vit: A light-weight classification model based on cnn and vit. In: ICIGP (2023)
44. Xiong, Y., Varadarajan, B., Wu, L., Xiang, X., Xiao, F., Zhu, C., Dai, X., Wang, D., Sun, F., Iandola, F., et al.: Efficientsam: Leveraged masked image pretraining for efficient segment anything. In: CVPR (2024)
45. Yang, J., Gao, M., Li, Z., Gao, S., Wang, F., Zheng, F.: Track anything: Segment anything meets videos. arXiv preprint arXiv:2304.11968 (2023)
46. Yang, L., Wei, Y.Z., He, Y., Sun, W., Huang, Z., Huang, H., Fan, H.: ishape: A first step towards irregular shape instance segmentation. arXiv preprint arXiv:2109.15068 (2021)
47. Yang, X., Ye, J., Wang, X.: Factorizing knowledge in neural networks. In: European Conference on Computer Vision. Springer (2022)
48. Yang, X., Zhou, D., Liu, S., Ye, J., Wang, X.: Deep model reassembly. NeurIPS **35**, 25739–25753 (2022)
49. Yu, F., Huang, K., Wang, M., Cheng, Y., Chu, W., Cui, L.: Width & depth pruning for vision transformers. In: AAAI (2022)
50. Yuan, Z., Xue, C., Chen, Y., Wu, Q., Sun, G.: Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In: ECCV (2022)
51. Zhang, C., Han, D., Qiao, Y., Kim, J.U., Bae, S.H., Lee, S., Hong, C.S.: Faster segment anything: Towards lightweight sam for mobile applications. arXiv preprint arXiv:2306.14289 (2023)
52. Zhang, C., Zhang, C., Li, C., Qiao, Y., Zheng, S., Dam, S.K., Zhang, M., Kim, J.U., Kim, S.T., Choi, J., et al.: One small step for generative ai, one giant leap for agi: A complete survey on chatgpt in aigc era. arXiv preprint arXiv:2304.06488 (2023)
53. Zhang, Q., Liu, X., Li, W., Chen, H., Liu, J., Hu, J., Xiong, Z., Yuan, C., Wang, Y.: Distilling semantic priors from sam to efficient image restoration models. In: CVPR (2024)
54. Zhao, X., Ding, W., An, Y., Du, Y., Yu, T., Li, M., Tang, M., Wang, J.: Fast segment anything. arXiv preprint arXiv:2306.12156 (2023)
55. Zwillinger, D., Kokoska, S.: CRC standard probability and statistics tables and formulae. Crc Press (1999)