

# Q-SAM2: Accurate Quantization for Segment Anything Model 2

Nicola Farronato<sup>◇♣</sup> Florian Scheidegger<sup>◇</sup> Mattia Rigotti<sup>◇</sup>

Cristiano Malossi<sup>◇</sup> Michele Magno<sup>♣</sup> Haotong Qin<sup>♣</sup>

<sup>◇</sup> IBM Research Zurich

<sup>♣</sup> ETH Zurich

nicola.farronato@ibm.com {eid, mrg, acm}@zurich.ibm.com  
{michele.magno, haotong.qin}@pbl.ee.ethz.ch

## Abstract

The Segment Anything Model 2 (SAM2) is a powerful foundation model for promptable segmentation. However, its high computational and memory costs are a major barrier to deployment on resource-constrained devices. In this paper, we present Q-SAM2, an accurate low-bit quantization method that achieves high compression and high fidelity. To address performance degradation arising from challenging weight and activation distributions during quantization, Q-SAM2 introduces two novel contributions: Variance-Reduced Calibration (VRC), an initialization method that reduces weight statistical variance by minimizing the Frobenius norm over a small calibration batch; and Learnable Statistical Clipping (LSC) a Quantization-Aware Training (QAT) method that learns momentum-stabilized clipping factors to manage outliers in weights and activations. Comprehensive experiments demonstrate that Q-SAM2 achieves highly accurate inference with substantial efficiency gains, significantly surpassing state-of-the-art general QAT schemes, particularly in the ultra-low 2-bit regime. Specifically, Q-SAM2 achieves an accuracy gain of up to 9.7 ppt in J&F on the video segmentation benchmark and 7.3 ppt in mIoU for instance segmentation over the best competing QAT model, all while achieving an 8x reduction in model size compared to the BF16 baseline.

## 1. Introduction

The Segment Anything Model 2 (SAM2) has emerged as the foundational framework for unified promptable segmentation across static images and video [32]. Its performance on video segmentation makes this model attractive for applications in robotics and tiny machine learning. However, taking advantage of real-time inference with this model re-

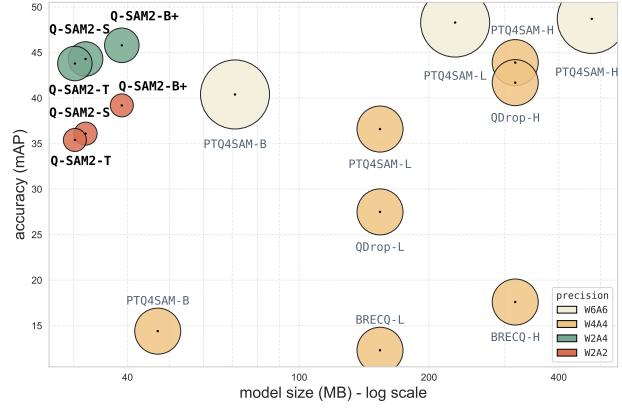


Figure 1. Comparison between Q-SAM2 and Post-Training Quantization (PTQ) based on SAM [18] [ViT-B/L/H]. Q-SAM2 defines a new SOTA Pareto frontier; even at its smallest precision (W2A2), it maintains accuracy comparable to higher-precision SOTA methods (PTQ4SAM [25], BRECQ [19], QDROP[34]).

quires high-end hardware, such as an NVIDIA A100 GPU. Furthermore, edge deployment scenarios typically impose tighter memory and storage constraints, even at the MB level in extreme cases, [13], which makes the use of the existing version of SAM2 prohibitive. A general effective strategy to compress and speed up deep neural networks is to quantize them using fewer bits. Quantization is a widely adopted technique for optimizing deep learning models, particularly when deploying them on resource-constrained hardware [12, 29]. By reducing the numerical precision of weights and activations, quantization enables faster inference, lower power, and smaller model sizes [3, 17].

For these reasons, the demand for post-training quantization (PTQ) methods has grown substantially in recent years, particularly due to the increasing size and complexity of foundation models like large language models

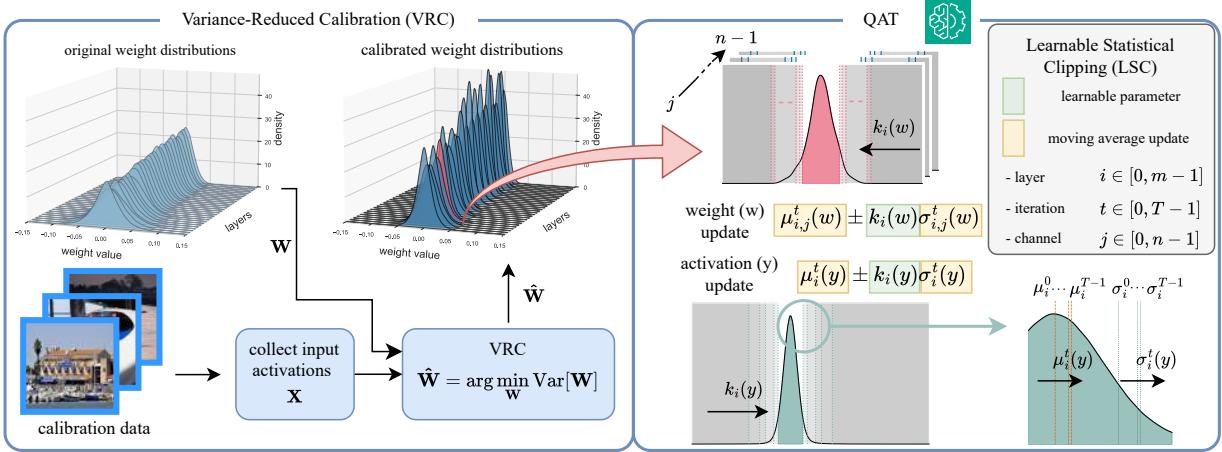


Figure 2. The Q-SAM2 approach. The weight distributions of the linear layers in the image encoder are calibrated using the VRC to reduce variance. We substitute the original encoder and train the network using the LSC in a QAT pipeline.

(LLMs) [1, 8, 11, 24, 25, 27, 36, 37]. These models often consist of billions of parameters, making them difficult to deploy efficiently without compression techniques. PTQ offers a practical solution by enabling model compression without requiring retraining, thus reducing computational costs and accelerating deployment [12, 19, 28]. Quantization Aware Training (QAT) [6, 7, 10, 15, 41], offers a more accurate solution by simulating quantization effects during the training phase, allowing the model to adapt to reduced numerical precision [17, 29, 35]. QAT enables the model to recover accuracy under aggressive compression, but its adoption remains limited in transformer architectures in resource-constrained scenarios due to its lack of extensive investigation and the high computational cost.

While no previous work has focused on the SAM2 architecture, recent studies have proposed PTQ algorithms for the original Segment Anything Model [18]. In particular, [23, 25] investigate the quantization challenges of SAM and introduce methods capable of producing per-channel quantized models with few data samples. Although PTQ avoids full retraining, the inability to adapt model parameters during quantization often leads to suboptimal results at ultra-low bit-widths. This limitation is highlighted in Figure 1, where state-of-the-art PTQ methods already exhibit significant performance degradation at 4-bit precision. The failure of PTQ at 4 bits motivates our work on a more robust, training-aware approach to achieve an accurate 2-bit model.

Quantizing SAM2 is nontrivial, particularly for the image encoder, a hierarchical transformer [33] which accounts for over 90% of the total parameters (in the B+ model) and poses a significant challenge for ultra-low-bit quantization (see Section 3). This difficulty stems from two main issues: the linear layer weight matrices exhibit values that deviate

significantly from the distribution center (introducing outliers), and the activation output displays heavy-tailed distributions. Both of these characteristics severely degrade performance when uniform quantization schemes are applied.

To address these challenges and achieve 2-bit precision, we propose Q-SAM2, a complete QAT pipeline. As shown in Figure 2, our approach introduces two novel components to tackle the critical issues of high weight variance and activation outliers: an initialization method called **Variance-Reduced Calibration (VRC)**, and a QAT algorithm named **Learnable Statistical Clipping (LSC)**. This two-step approach first prepares the model for quantization and then robustly trains it, ensuring high accuracy even at 2-bit precision. Our main contributions are as follows:

- We introduce Q-SAM2, the **first** quantization pipeline for the SAM2 architecture. Our method establishes a new state-of-the-art for QAT, surpassing the stronger baseline up to 7.3ppt in mIoU. We provide validation of results against SOTA PTQ methods and demonstrate robust accuracy even in ultra-low bit schemes.
- We propose a novel light calibration method, VRC, that reduces the variance of the encoder’s weight distributions using a small batch of images. This procedure significantly lowers the initial quantization error, providing an improved starting point for the subsequent training.
- We develop a tailored QAT pipeline that leverages the calibrated encoder (from VRC) and a novel algorithm, LSC. The LSC grounds the quantization range using robust EMA-tracked statistical estimates ( $\mu, \sigma$ ). This hybrid design introduces only a single, learnable coefficient ( $k$ ) per quantizer, enabling stable ultra-low-bit convergence.

The paper is organized as follows: Section 2 describes the preliminaries and related work, Section 3 shows the Q-

SAM2 pipeline, Section 4 presents our experimental setup and results, and Section 5 concludes our findings.

## 2. Preliminaries and Related Work

### 2.1. Model Quantization

Quantization is the process of approximating a continuous or high-resolution discrete domain by a finite set of representative levels, to optimize computational efficiency and memory usage with controlled performance degradation. The quantization process can be formulated as:

$$\mathbf{X}_q = \text{clip} \left( \text{round} \left( \frac{\mathbf{X}}{s} \right) + z, 0, 2^b - 1 \right). \quad (1)$$

Where  $\mathbf{X}$  is a high-precision input tensor (e.g., FP32). The process maps  $\mathbf{X}$  to the low-bit integer tensor  $\mathbf{X}_q$  using the scaling factor  $s$  and the integer zero-point  $z$ . Specifically,  $s$  defines the step size of the quantization levels, while  $z$  acts as an offset ensuring that the real-valued zero is mapped exactly to an integer. The bit-width  $b$  determines the cardinality of the discrete set, typically  $[0, 2^b - 1]$  for unsigned quantization. The clipping function is mathematically necessary to constrain the projected values  $\text{round}(\mathbf{X}/s) + z$  to this finite integer domain. These parameters can be defined as scalars ( $s, z \in \mathbb{R}$ ) for per-tensor quantization, or vectors ( $\mathbf{s}, \mathbf{z} \in \mathbb{R}^d$ ) for per-channel quantization or a tensor ( $\mathbf{s}, \mathbf{z} \in \mathbb{R}^{d_1 \times d_2 \times \dots}$ ) for per-group quantization. We focus on per-channel uniform quantization, which offers a good balance between efficiency and complexity.

### 2.2. Post-Training Quantization (PTQ)

Post-training quantization (PTQ) enables efficient deployment of large pretrained models by applying quantization without retraining the full network. Scale  $s$  and zero-point  $z$  are typically computed from calibration data. To improve accuracy, various PTQ methods introduce transformations to reduce quantization error: AdaRound [28] and BrecQ [19] optimize rounding and reconstruction, respectively. QDROP [34] further refined this by adding a Dropout-inspired regularization, making weight optimization more robust to activation quantization errors. Other approaches leverage Hessian-based and calibration-free formulations like GPTQ [11] and HQQ [2], or minimizing weight outliers via  $L_\infty$ -based regularization as in MagR [39]. Activation outliers, common in transformers [8, 36], further complicate quantization. Techniques such as SmoothQuant [37], SpinQuant [24], and QuaRot [1] mitigate this by shifting or rotating activations to promote accurate PTQ.

In the vision domain, PTQ4ViT [38] tackles non-Gaussian activations in vision transformers. PTQ4DiT [36] introduces channel-wise salience balancing to mitigate salient-channel and temporal activation

outliers in Diffusion Transformers. PTQ4SAM [25] is the first PTQ method tailored to SAM, utilizing bimodal integration and adaptive granularity quantization to transform challenging activation distributions.

### 2.3. Quantization Aware Training (QAT)

In Quantization Aware Training (QAT), the quantization process is simulated during training through Equation 1, allowing the network to adapt to quantization noise and minimize accuracy degradation. The scale  $s$  and zero point  $z$  can be treated as learnable parameters or are dynamically adapted, and the model weights themselves are updated through backpropagation to account for quantization effects. This joint adaptation allows the model to recover from quantization-induced distortions and is essential to preserve accuracy under low-bit regimes. Formally, QAT introduces non-differentiable quantization into the training loop, typically handled using straight-through estimators (STE) [4] to enable gradient-based optimization.

Several QAT methods have been proposed to improve accuracy in low-precision regimes. One of the first and most influential works is the MinMax-based QAT approach [17], which uses affine quantization and simulated integer operations during training, enabling efficient 8-bit deployment with integer-only inference. Another effective method is DoReFa-Net [41], which introduces quantization of weights, activations, and gradients using deterministic functions. However, its use in modern transformer architectures is limited due to the inability to quantize models with large activation ranges [20]. PACT [7] introduced a learnable clipping parameter for activations, enabling better control of dynamic range. Nevertheless, this introduces additional parameters and gradients, complicating the deployment. LSQ [10] is widely regarded as one of the most robust QAT methods, learning the quantization step size via backpropagation. Building on this, LSQ+ [5] improves training stability by introducing a log-based gradient formulation for the learnable scale, which prevents the scale from becoming zero and stabilizes the training process.

QAT remains underexplored in transformer-based vision architectures, where challenges such as weight oscillation, activation outliers, and attention-specific sensitivity can severely impact low-bit performance. Existing studies [14, 22] focus solely on LSQ, without evaluating alternative strategies or providing broader comparisons between quantization methods.

## 3. Method

### 3.1. Challenges in Quantizing SAM2

The SAM2 architecture is built on a Hiera-based encoder [33], which hierarchically merges tokens for an efficient multiscale representation. While effective for dense

vision tasks, this architecture introduces challenges for quantization due to the presence of outliers in both weights and activations. We find that weight distributions are heavy-tailed, with extreme values concentrated in early layers, leading to instability under low-bit quantization. Activation outliers, though sparse, consistently appear in specific channels, further degrading precision. Together, these factors cause significant performance drops when reducing weight precision to 4 or even 2 bits.

As a preliminary investigation, we analyze explicit value clipping on weight matrices to mitigate outliers by truncating extreme values to increase resolution in dense regions. While this results in an initial accuracy drop, the preservation of core predictive behavior indicates that these outliers are non-essential for effective functioning. We observe the same behavior for activations, which are even more difficult to quantize on low bits because of their higher value range. The complete findings and supporting analysis for this preliminary investigation are detailed in Appendix 6.3. Our method is built on a key insight: since outliers can be clipped without destroying the model’s predictive power, the problem becomes optimizing the remaining, dense part of the distribution. We therefore propose a two-stage pipeline: first, we apply **Variance-Reduced Calibration (VRC)** (Section 3.2), a novel initialization that narrows the encoder’s weight distributions. This provides a low variance, so a lower quantization error starting point. Second, we apply **Learnable Statistical Clipping (LSC)** (Section 3.3), a robust QAT algorithm that learns optimal clipping thresholds for both calibrated weights and activations.

### 3.2. Variance-Reduced Calibration (VRC)

Although weight calibration is common in PTQ, the idea of introducing a dedicated preprocessing step before QAT has received little attention [20]. In this work, we show that VRC can improve performance, particularly in ultra-low bit-width regimes.

Designed as a precursor to our clipping-based QAT, our calibration minimizes weight variance to compact the distribution. This allows for a smaller quantization step size, directly reducing the quantization error for the majority of the weights. We employ a computationally efficient, closed-form minimization of the Frobenius norm using a small calibration set. This approach is theoretically grounded: since the weight means are generally close to zero, the Frobenius norm serves as an accurate proxy for variance.

However, any modification of a fully trained linear layer weight matrix risks degrading the accuracy of the model. A straightforward but crucial observation is that the accuracy degradation in the final model is directly related to the discrepancy between the output of the linear layers before and after quantization, rather than the quantization error of the matrix elements themselves. Any changes of  $\mathbf{W}$  will there-

fore need to balance the compression gains due to quantization with the resulting errors at the output of linear layers.

Based on this insight, we propose a quantization procedure aimed at optimizing this trade-off through an initial calibration step that projects the trained weights onto the subspace spanned by a calibration batch of data samples, which ensures that the layer’s outputs are preserved on these known inputs. In particular, we propose to perform this projection learning the *Moore-Penrose Pseudoinverse* between the input activations and the output of the linear layer while running a forward pass of the original trained model on the calibration samples. The use of the Moore-Penrose pseudoinverse is motivated by its known properties in terms of 1) minimizing the residual with the calibration output activations and 2) minimizing the Frobenius norm of the obtained weights. In other words, this simultaneously achieves that the dynamic range of the weight distribution is minimized, making them easier to quantize, while at the same time minimizing the change of output of the linear layer (at least on the calibration samples).

While the residual between pre- and post-calibration output activations is guaranteed to be minimized only on the calibration samples, the Moore-Penrose Pseudoinverse has additional properties that make it suitable as a preprocessing step before quantization. The fact that it allows us to control the singular values of the calibrated weight matrix (see later the discussion on Tikhonov regularization) implies that the calibration process “cleans” its spectrum by eliminating components that are irrelevant, unstable, or poorly supported by the calibration data, leading to a simpler and more robust solution. As a result, the modified layer maintains essential functionality on the calibration inputs while promoting smoother and potentially more general behavior elsewhere, which is particularly important for robustness under quantization or other perturbations.

The selection of calibration data is critical: larger batches improve faithfulness to the original model, while smaller ones reduce memory and runtime costs. Moreover, choosing representative frames is equally important because stacking ill-conditioned inputs in the calibration set can collapse the key directions learned during training, harming generalization. As a result, the layer may perform well on calibration samples, but fail on unseen or diverse data. Henceforth, calibration inputs must be diverse, well-conditioned, and representative of the operational domain of the model.

The condition number of activations reflects the calibration input quality: low values indicate stable and rich subspaces. However, aggregating inputs presents a risk: a single ill-conditioned sample can propagate instability to the entire batch, compromising the overall projection quality. Since pre-filtering inputs is computationally infeasible, we employ regularization to suppress ill-conditioning, enabling

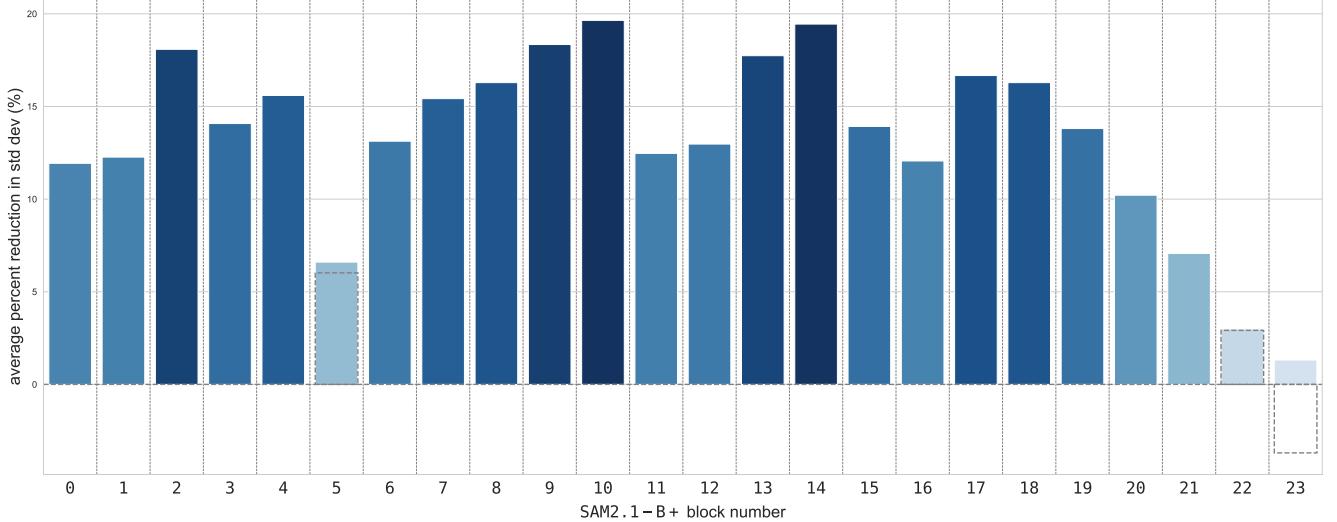


Figure 3. Impact of VRC on SAM2.1-B+ image encoder weight distributions for  $\lambda_0 = 2.0$ . VRC achieves a significant (10-20%) reduction in standard deviation, averaged per transformer block. This compresses the dynamic range, critically lowering the initial quantization error for the subsequent QAT.

random input selection.

A linear layer is defined by  $f(\mathbf{X}, \mathbf{W})$ . It maps an input tensor  $\mathbf{X} \in \mathbb{R}^{n \times d_{\text{in}}}$ , where  $n$  is the batch size and  $d_{\text{in}}$  is the input feature dimension, to an output tensor  $\mathbf{Y} \in \mathbb{R}^{n \times d_{\text{out}}}$ , where  $d_{\text{out}}$  is the output feature dimension, through an affine transformation defined as  $f(\mathbf{X}, \mathbf{W}) = \mathbf{X}\mathbf{W}^\top + \mathbf{b}$ , with  $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$  and  $\mathbf{b} \in \mathbb{R}^{d_{\text{out}}}$ . We consider  $\mathbf{b} = \mathbf{0}$  for the following analysis, without loss of generality.

We compute the minimum Frobenius norm weight matrix  $\hat{\mathbf{W}}$  for the batch  $\mathbf{X}$  using the Moore-Penrose pseudoinverse [30]. For  $\mathbf{X} \in \mathbb{R}^{n \times d}$  and  $\mathbf{Y} \in \mathbb{R}^{n \times p}$ , among all matrices  $\mathbf{W} \in \mathbb{R}^{p \times d}$  satisfying  $\mathbf{Y} = \mathbf{X}\mathbf{W}^\top$ , the unique matrix minimizing the Frobenius norm  $\|\mathbf{W}\|_F$  is

$$\hat{\mathbf{W}}^\top = \mathbf{X}^\dagger \mathbf{Y}, \quad (2)$$

where  $\mathbf{X}^\dagger$  denotes the Moore-Penrose pseudoinverse of  $\mathbf{X}$ .

To mitigate the instability caused by poorly conditioned calibration samples, we applied the Tikhonov regularization during the projection. From Equation (2) the new formulation is derived:

$$\hat{\mathbf{W}}^\top = (\mathbf{X}^\top \mathbf{X} + \lambda I)^{-1} \mathbf{X}^\top \mathbf{Y} \quad (3)$$

where  $I$  is the  $d \times d$  identity matrix, and  $\lambda \in \mathbb{R}^+$  is the cutoff hyperparameter. We set  $\lambda$  based on the singular values of the activation matrix to ensure stability during pseudoinverse computation. When the matrix is ill-conditioned but not rank-deficient,  $\lambda$  is chosen as a small multiple of the smallest non-zero singular value. In cases where the smallest singular value indicates an effective rank deficiency (e.g., in low-precision formats like `bfloat16`), we instead select the smallest stable singular value and scale

it accordingly. In practice,  $\lambda = \lambda_0 \sigma_*$ , where  $\sigma_*$  is the smallest singular value that avoids rank deficiency, and  $\lambda_0$  ranges between one and five. In Appendix 6.1 we present an analysis of our calibration method and the role of  $\lambda$  in Equation 3. Figure 3 quantifies the efficacy of our VRC method on the SAM2.1-B+ model (using  $\lambda_0 = 2.0$ ). The chart plots the average percent reduction in standard deviation across the linear layers within each transformer block. The dotted blocks contain layers with negative gains due to high instability in the input activations, in which the selected  $\lambda_0$  is not sufficient to reduce the variance. In this case, the algorithm applies a fallback to the original layer, resulting in the normal colored block. Most blocks achieve a significant reduction of more than 10%, with some approaching 20%. By systematically reducing the weight variance, the VRC compresses the dynamic range of the distributions. This reduction in the initial quantization error is critical for providing a stable and low-error starting point for the subsequent QAT phase. We report the calibrated encoder layer-by-layer in Appendix 6.1.

### 3.3. Learnable Statistical Clipping (LSC)

Although learnable QAT methods like LSQ [10] and LSQ+ [5] are effective, their “free-floating” learnable parameters (e.g., step-size  $s$  and offset  $z$ ) are not statistically grounded. This is a known source of instability, particularly in the per-channel, low-bit regime, where the large number of sensitive parameters leads to instability or variance, requiring a great deal of hyper-parameter tuning and initialization [5].

To solve this, we introduce **Learnable Statistical Clipping (LSC)**, a hybrid method that replaces these unstable

learnable parameters with robust, statistically-grounded estimates. LSC anchors the quantization bounds to the data’s true distribution, leaving only a single, well-behaved parameter to be optimized for each quantizer.

First, we track the first two moments of the distribution using an Exponential Moving Average (EMA), analogous to the robust estimators in Batch Normalization [16]. For weights, we track, on each layer, the per-channel batch statistics  $\hat{\mu}_{w,c}^{(t)}$  and  $\hat{\sigma}_{w,c}^{(t)}$  at iteration  $t$ :

$$\begin{aligned}\mu_{w,c}^{(t)} &= \beta \mu_{w,c}^{(t-1)} + (1 - \beta) \hat{\mu}_{w,c}^{(t)}, \\ \sigma_{w,c}^{(t)} &= \beta \sigma_{w,c}^{(t-1)} + (1 - \beta) \hat{\sigma}_{w,c}^{(t)},\end{aligned}$$

where  $\beta \in [0, 1]$  is the momentum term. We apply the same statistical tracking principle at the per-tensor level for activations.

The impact of this temporal smoothing is different for the two components. Activation batch statistics  $(\hat{\mu}_a^{(t)}, \hat{\sigma}_a^{(t)})$  are inherently noisy, as they are highly dependent on the input data. The momentum  $\beta$  is therefore essential for providing a stable, time-averaged estimate of the activation distribution. In contrast, weight statistics are much more static, especially as training converges. While the EMA still provides stabilization, its primary role for weights is to establish a robust estimate during the initial phase of QAT.

Second, we introduce a single, learnable, tensor-wide parameter  $k$  for each quantizer. This learnable  $k$  is optimized via backpropagation to find the optimal clipping range. This allows the network to learn the best balance between clipping error (the information lost from truncating outliers, which dominates when  $k$  is small) and quantization error (the loss of precision from using wide buckets, which dominates when  $k$  is large).

The final dynamic ranges of the quantized values for weights, as presented in Equation 1, are then defined symmetrically around the tracked mean:

$$\begin{aligned}\min_{w,c}^{(t)} &= \mu_{w,c}^{(t)} - |k_w| \sigma_{w,c}^{(t)}, \\ \max_{w,c}^{(t)} &= \mu_{w,c}^{(t)} + |k_w| \sigma_{w,c}^{(t)}.\end{aligned}$$

The per-tensor activation bounds  $(\min_a^{(t)}, \max_a^{(t)})$  are computed analogously using their respective per-tensor statistics and learnable coefficient  $k_a$ .

This formulation is fundamentally more stable. A single layer in LSQ+ requires optimizing thousands of sensitive, independent per-channel parameters  $(s_c, z_c)$  for weights alone. In contrast, our LSC replaces this entire set with robust, non-learnable statistics  $(\mu_{w,c}^{(t)}, \sigma_{w,c}^{(t)})$  and just one learnable scalar,  $k_w$ . The optimization landscape is thus dramatically simplified. The gradients for  $k_w$  and  $k_a$  are well-behaved sums aggregated across all channels or tensor elements and are mediated by the stable  $\sigma^{(t)}$  statistics.

## 4. Experiments

### 4.1. Experimental Setup

#### 4.1.1. Task, Dataset, and Metrics

Our experimental setup resembles the one in the original SAM2 work [32]. We apply the VRC and train the models on a subset of the SA-1B [18] and SA-V [32] datasets. The SA-1B dataset is composed of 11M images and more than 1B masks, while SA-V contains 51k videos and 643K spatio-temporal segmentation masks. To reduce training time, we use a smaller subset of the dataset and still obtained high accuracy, demonstrating that we achieve performance without full retraining. From the original data, we subsample at random  $\sim 8\%$  and  $\sim 45\%$ , respectively.

We evaluate our models on the semi-supervised Visual Object Segmentation (VOS) task, which involves segmenting a target object across a video given its ground-truth mask in the first frame. Following the SAM2 evaluation setup, we use SA-V val, SA-V test (both subsets of SA-V), and MOSE val from the MOSE dataset [9], reporting  $J\&F$  accuracy [31]. For instance segmentation, we evaluate on the COCO2017 validation set [21], using mean Intersection over Union (mIoU) across all 5,000 images. Moreover, following the same evaluation as PTQ4SAM [25], we use the predicted boxes generated from DINO [40] as input prompt for our model, in order to calculate the mean Average Precision (mAP) on COCO2017.

#### 4.1.2. Implementation

Our starting models are the checkpoints available from the SAM2 repository [26], named sam2.1\_hiera\_tiny (T), sam2.1\_hiera\_small (S), sam2.1\_hiera\_base\_plus (B+). The implementation of the QAT pipelines is based on PyTorch 2.6.0 and CUDA 12.4, and all experiments are performed on NVIDIA A100 80GB GPUs. The VRC is then performed using 50 input images, chosen uniformly at random from the SA-1B dataset. We forward the images through the original image encoder and record the activations. After stacking the tensors, we obtain the calibrated weight matrices by applying equation 3 with  $\lambda_0 = 2.0$ . To assess our method, we compare Q-SAM2 against the classical MinMax method [17], PACT [7], and LSQ+ [5]. Since standard LSQ+ (and LSQ [10]) fails to converge with per-channel quantization [20], we implement a Tuned LSQ+ (T.LSQ+) baseline. This involves restricting the original LSQ+ to only activations quantization, tuning the learning rate for the step size  $s$  and offset  $z$ , increasing observer calibration samplings, and employing per-channel MinMax for weights. To ensure a fair comparison, we evaluate LSQ+ against Q-SAM2 in the W4A4 per-tensor schema. Even in this setting, Q-SAM2 maintains a significant lead, outperforming LSQ+ by 9.7 ppt (VOS J&F) and 7.7 ppt

Metric	Dataset	Methods	SAM2.1-B+			SAM2.1-S			SAM2.1-T		
			FP	W2A4	W2A2	FP	W2A4	W2A2	FP	W2A4	W2A2
<i>J &amp; F</i>	SA-V Val [32]	MinMax [17]		52.3	18.7		56.8	27.8		53.5	17.9
		PACT [7]	78.1	54.7	36.4	77.7	58.4	40.0	75.0	56.3	43.4
		T_LSQ+ [5]		55.0	50.7		58.1	51.7		57.5	51.4
		<b>Q-SAM</b>		<b>64.7</b>	<b>55.2</b>		<b>65.1</b>	<b>56.0</b>		<b>64.1</b>	<b>54.8</b>
mAP	MOSE Val [9]	MinMax [17]		56.8	21.2		59.9	29.7		57.1	19.5
		PACT [7]	78.2	59.0	37.4	76.6	60.6	43.1	75.0	59.5	45.5
		T_LSQ+ [5]		59.0	53.3		60.8	54.3		60.0	55.5
		<b>Q-SAM2</b>		<b>64.3</b>	<b>58.3</b>		<b>67.9</b>	<b>59.1</b>		<b>67.1</b>	<b>59.3</b>
mIoU	COCO2017 [21]	MinMax [17]		55.3	26.5		59.1	33.5		58.8	26.2
		PACT [7]	73.7	58.6	40.8	73.5	60.6	43.7	70.9	60.9	44.3
		T_LSQ+ [5]		56.0	52.9		61.5	53.0		60.2	54.7
		<b>Q-SAM2</b>		<b>64.1</b>	<b>57.1</b>		<b>67.4</b>	<b>57.6</b>		<b>66.8</b>	<b>58.2</b>
		MinMax [17]		36.2	15.4		37.5	20.3		36.5	16.8
		PACT [7]	49.6	38.3	25.2	48.4	38.8	27.2	48.0	38.7	28.2
		T_LSQ+ [5]		39.3	34.6		40.2	31.4		38.8	32.3
		<b>Q-SAM2</b>		<b>45.8</b>	<b>39.2</b>		<b>44.3</b>	<b>36.1</b>		<b>43.8</b>	<b>35.4</b>
		MinMax [17]		39.0	25.0		41.2	26.9		40.3	26.5
		PACT [7]	59.1	41.2	32.0	59.7	41.9	32.6	57.1	38.7	28.2
		T_LSQ+ [5]		42.1	37.9		42.6	36.4		42.5	37.3
		<b>Q-SAM2</b>		<b>49.4</b>	<b>42.1</b>		<b>47.3</b>	<b>39.6</b>		<b>47.5</b>	<b>39.8</b>

Table 1. Quantitative results on Video Object Segmentation (VOS) and Instance Segmentation. Our method is highlighted in grey. **Bold** indicates the best result for each configuration.

(COCO mIoU). Full details, training pipelines, parameters, and losses are provided in Appendix 6.2. Finally, we validate the stability of our training pipeline, observing a marginal standard deviation of 0.26 ppt in mIoU across independent runs.

#### 4.2. Ultra-Low-Bit Quantization Performance

Table 1 summarizes the results of our experiments. The results show that our method consistently outperforms the other QAT algorithms by a large margin across each task, dataset, encoder size, and precision configuration. For the B+ encoder, Q-SAM2 demonstrates substantial improvement over the best baseline, achieving average gains on the video benchmark of 7.0 ppt (W2A4) and 4.6 ppt (W2A2). For instance segmentation, gains are 6.5 ppt (W2A4) and 4.6 ppt (W2A2) in mAP, and 7.3 ppt (W2A4) and 4.2 ppt (W2A2) in mIoU. The performance margin is reduced for the small and tiny encoders. This narrowing is expected because smaller models are inherently easier to quantize due to fewer outliers and narrower weight and activation distributions compared to the B+ configuration. This observation is attributed to the fact that the simpler distributions of the S and T encoders significantly boost the absolute performance of existing QAT baselines, thereby narrowing the gap with Q-SAM2. While T\_LSQ+ substantially reduces the performance difference in the aggressive W2A2 configuration, Q-SAM2 retains a significant lead. While the tiny configuration yields our smallest performance margin rel-

ative to baselines, representing the worst-case scenario for our gains, we still achieve substantial improvements averaging 3.6 ppt on the video benchmark, 3.1 ppt in mAP, and 2.5 ppt in mIoU against T\_LSQ+.

In a second experiment we evaluate our Q-SAM2 against post-training quantization methods, specifically PTQ4SAM [25], BRECQ [19], and QDROP [34], on the instance segmentation task. As visualized in Figure 1, our lightweight Q-SAM2 W2A4\_B+ model achieves a 1.9 ppt mAP gain over the more than  $\times 8$  larger (in MB) PTQ4SAM-H W4A4 model. The complete comparison is reported in Appendix 6.2.

#### 4.3. Ablation Studies

Table 2. Ablation study on VRC and LSC for B+ model. We report mAP for instance segmentation on COCO2017 dataset.

Dataset	VRC ( $\lambda_0 = 2.0$ )	LSC	W2A2
COCO2017 [21]	✗	✗	15.4
	✓	✗	18.3
	✗	✓	37.1
	✓	✓	<b>39.2</b>

Table 2 reports the contributions of VRC and LSC in trained networks. While the largest contribution stems from the LSC’s management of activation outliers, VRC yields a 2.1 ppt gain. This improvement is statistically significant,



Figure 4. Qualitative results on W2A2 configuration for B+ encoder on promptable instance segmentation task.

Table 3. Ablation studies on the VRC for weight-only quantization for B+ encoder. The mAP for the image segmentation on COCO2017 and J&F for the VOS on SA-V val are reported.

Dataset	Method	VRC ( $\lambda_0$ )	W3	W4
COCO2017 [21]	HQQ [2]	✗	24.9	47.5
		✓ (1.5)	29.4	<b>47.6</b>
		✓ (2.0)	<b>29.6</b>	46.0
	MinMax [17]	✗	15.7	46.3
		✓ (1.5)	19.6	<b>46.9</b>
		✓ (2.0)	<b>20.3</b>	45.2
SA-V val [32]	HQQ [2]	✗	38.1	<b>71.0</b>
		✓ (1.5)	41.2	<b>71.0</b>
		✓ (2.0)	<b>43.0</b>	67.6
	MinMax [17]	✗	34.4	69.2
		✓ (1.5)	36.6	<b>70.9</b>
		✓ (2.0)	<b>39.5</b>	64.4

as it substantially exceeds the observed training standard deviation of 0.26 ppt. The second ablation study evaluates the VRC calibration procedure in a post-training quantization setting using the B+ model with 3- and 4-bit weight-only quantization. We test VRC with two values of  $\lambda_0$  on two methods: standard MinMax [17] and the Hessian-aware HQQ [2]. As detailed in Table 3, VRC consistently improves performance across both strategies. The gain is most significant in the 3-bit configuration, achieving a 4.6 ppt mAP gain for MinMax. In the 4-bit setting, the improve-

ment is marginal ( $\lambda_0 = 1.5$  offers a slight gain), confirming the expected trend: the contribution of VRC decreases as the bit-width becomes sufficient to represent weights accurately.

#### 4.4. Qualitative Results

Figure 4 presents a qualitative comparison of Q-SAM2 versus full-precision SAM2.1 [32] and quantized baselines PACT [7] and T-LSQ+ [5]. Visually confirming our quantitative advantage, Q-SAM2 demonstrates markedly improved segmentation quality. PACT consistently exhibits high levels of artifacts across all images and substantially fails to segment the input prompt. In contrast, T-LSQ+ fares marginally better, achieving only partial segmentation in some instances but invariably introducing noticeable artifacts. Our Q-SAM2 yields segmentations closest to the full-precision model. Additional visual results and a video example are available in Appendix 6.4.

#### 5. Conclusions

We presented Q-SAM2, to the best of our knowledge the first quantized Segment Anything Model 2. Our method combined VRC, which reduces weight variance, and LSC, a robust QAT procedure that learns to clip outliers. Q-SAM2 significantly outperformed strong baselines on VOS and instance segmentation. Across all encoder sizes and quantization schemes evaluated, we observed average gains of

5.5 ppt J&F on the video benchmark, 4.7 ppt in mAP, and 4.5 ppt in mIoU, while qualitative results confirmed its effectiveness in preserving segmentation quality.

Future work could investigate the generalization of VRC and LSC to other architectures. A particularly promising direction involves exploring an adaptive VRC with a layer-specific  $\lambda$  parameter, which enables a more fine-grained calibration to further optimize the accuracy-compression trade-off.

## Acknowledgement

We acknowledge funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie actions HORIZON-MSCA-2022-DN-01 call (Grant agreement ID: 101119554).

## References

- [1] Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Martin Jaggi, Dan Alistarh, Torsten Hoefer, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. *arXiv preprint arXiv:2404.00456*, 2024. [2](#) [3](#)
- [2] Hicham Badri and Appu Shaji. Half-quadratic quantization of large machine learning models, 2023. [3](#) [8](#)
- [3] Ron Banner, Yury Nahshan, and Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. *Advances in neural information processing systems*, 32, 2019. [1](#)
- [4] Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013. [3](#)
- [5] Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. Lsq+: Improving low-bit quantization through learnable offsets and better initialization, 2020. [3](#) [5](#) [6](#) [7](#) [8](#) [11](#) [12](#) [14](#) [17](#)
- [6] Zhaowei Cai, Xiaodong He, Jian Sun, and Nuno Vasconcelos. Deep learning with low precision by half-wave gaussian quantization. In *2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5406–5414, 2017. [2](#)
- [7] Y. Choi, M. El-Khamy, and J. Lee. Pact: Parameterized clipping activation for quantized neural networks. In *arXiv preprint arXiv:1805.06085*, 2018. [2](#) [3](#) [6](#) [7](#) [8](#) [12](#)
- [8] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2022. Curran Associates Inc. [2](#) [3](#)
- [9] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip HS Torr, and Song Bai. MOSE: A new dataset for video object segmentation in complex scenes. In *ICCV*, 2023. [6](#) [7](#) [14](#)
- [10] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. In *International Conference on Learning Representations (ICLR)*, 2019. [2](#) [3](#) [5](#) [6](#) [14](#)
- [11] Elias Frantar, Saleh Ashkboos, Torsten Hoefer, and Dan Alistarh. GPTQ: Accurate post-training compression for generative pretrained transformers. *arXiv preprint arXiv:2210.17323*, 2022. [2](#) [3](#)
- [12] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-power computer vision*, pages 291–326. Chapman and Hall/CRC, 2022. [1](#) [2](#)
- [13] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, 2016. [1](#)
- [14] Xijie Huang, Zhiqiang Shen, Pingcheng Dong, and Kwang-Ting Cheng. Quantization variation: A new perspective on training transformers with low-bit precision. *Transactions on Machine Learning Research*, 2024. [3](#)
- [15] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: training neural networks with low precision weights and activations. *J. Mach. Learn. Res.*, 18(1):6869–6898, 2017. [2](#)
- [16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. [6](#)
- [17] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018. [1](#) [2](#) [3](#) [6](#) [7](#) [8](#) [12](#)
- [18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. [1](#) [2](#) [6](#) [11](#) [15](#) [17](#)
- [19] Jianfei Li, Shupeng Zhang, Kaiqi Liu, and Zhen Han. Brecq: Pushing the limit of post-training quantization by block reconstruction. *arXiv preprint arXiv:2102.05426*, 2021. [1](#) [2](#) [3](#) [7](#) [15](#) [17](#)
- [20] Yuhang Li, Mingzhu Shen, Jian Ma, Yan Ren, Mingxin Zhao, Qi Zhang, Ruihao Gong, Fengwei Yu, and Junjie Yan. Mqbench: Towards reproducible and deployable model quantization benchmark, 2022. [3](#) [4](#) [6](#) [14](#)
- [21] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. [6](#) [7](#) [8](#) [14](#) [17](#)
- [22] Shih-Yang Liu, Zechun Liu, and Kwang-Ting Cheng. Oscillation-free quantization for low-bit vision transformers. In *Proceedings of the 40th International Conference on Machine Learning*, pages 21813–21824. PMLR, 2023. [3](#)
- [23] Xiaoyu Liu, Xin Ding, Lei Yu, Yuanyuan Xi, Wei Li, Zhi-jun Tu, Jie Hu, Hanting Chen, Baoqun Yin, and Zhiwei

- Xiong. Pq-sam: Post-training quantization for segment anything model. In *Computer Vision – ECCV 2024*, pages 420–437, Cham, 2025. Springer Nature Switzerland. 2
- [24] Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations, 2025. 2, 3
- [25] Chengtao Lv, Hong Chen, Jinyang Guo, Jinyang Guo, Jinyang Guo, Yifu Ding, Xianglong Liu, Xianglong Liu, and Xianglong Liu. Ptq4sam: Post-training quantization for segment anything. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15941–15951, 2024. 1, 2, 3, 6, 7, 15, 17
- [26] Meta AI Research. SAM2 github repository. <https://github.com/facebookresearch/sam2>, 2024. Accessed November 25, 2025. 6
- [27] Markus Nagel, Mart Van Baalen, Tijmen Blankevoort, and Max Welling. Data-Free Quantization Through Weight Equalization and Bias Correction . In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1325–1334, Los Alamitos, CA, USA, 2019. IEEE Computer Society. 2
- [28] Markus Nagel, Mart van Baalen, Titus Blankevoort, and Max Welling. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning (ICML)*, pages 7197–7206, 2020. 2, 3
- [29] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart Van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021. 1, 2
- [30] R. Penrose. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413, 1955. 5
- [31] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *CoRR*, abs/1704.00675, 2017. 6
- [32] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädlé, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1, 6, 7, 8, 11, 12, 14
- [33] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, Jitendra Malik, Yanghao Li, and Christoph Feichtenhofer. Hiera: a hierarchical vision transformer without the bells-and-whistles. In *Proceedings of the 40th International Conference on Machine Learning*. JMLR.org, 2023. 2, 3
- [34] Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. Qdrop: Randomly dropping quantization for extremely low-bit post-training quantization, 2023. 1, 3, 7, 15, 17
- [35] Hao Wu, Patrick Judd, Xiaojie Zhang, Mikhail Isaev, and Paulius Micikevicius. Integer quantization for deep learning inference: Principles and empirical evaluation. *arXiv preprint arXiv:2004.09602*, 2020. 2
- [36] Junyi Wu, Haoxuan Wang, Yuzhang Shang, Mubarak Shah, and Yan Yan. Ptq4dit: Post-training quantization for diffusion transformers. In *NeurIPS*, 2024. 2, 3
- [37] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. 2, 3
- [38] Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *Computer Vision – ECCV 2022*, pages 191–207, Cham, 2022. Springer Nature Switzerland. 3
- [39] Aozhong Zhang, Naigang Wang, Yanxia Deng, Xin Li, Zi Yang, and Penghang Yin. Magr: Weight magnitude reduction for enhancing post-training quantization. *arXiv preprint arXiv:2406.00800*, 2024. 3
- [40] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection, 2022. 6
- [41] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, and He Wen. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. In *arXiv preprint arXiv:1606.06160*, 2016. 2, 3

## 6. Technical Appendices and Supplementary Material

This appendix provides more details to support the findings presented in the main paper. In Section 6.1, we analyze the bias-variance trade-off introduced by the regularization hyperparameter  $\lambda$  in the VRC, Section 6.2 outlines our training configuration and other results. In particular Section 6.2.2 presents training loss curves, and Section 6.2.3 discusses the differences between our training procedure and the original SAM2 model [32]. Section 6.2.4 the LSQ+ [5] QAT algorithm is compared with our Q-SAM2 in a per-tensor schema, and in Section 6.2.5 we compare our performance with PTQ algorithm applied on SAM [18]. We present insights on the weight and activation distributions of SAM2 in Section 6.3. Finally, Section 6.4 includes qualitative segmentation results to illustrate the effectiveness of our approach in both static images and a video.

### 6.1. Variance-Reduced Calibration Details

To better analyze the behavior of our calibration method introduced in Section 3.2, we implement Equation 3 within the linear layer `cut.blocks.21.mlp.layers.1` of the image encoder in the B+ model. Calibration is performed by applying the regularized pseudoinverse formulation to batches of 50 images, chosen uniformly at random, while systematically varying the regularization hyperparameter  $\lambda$ . For each value of  $\lambda$ , we compute a calibrated weight matrix  $\hat{\mathbf{W}}_\lambda \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ .

To evaluate the effect of calibration, we compare the outputs of the original and calibrated layers using a separate evaluation batch of 10 images, denoted as  $\mathbf{X}_{\text{eval}}$ . The output discrepancy is quantified using the  $L_2$  norm between the original projection  $\mathbf{X}_{\text{eval}} \mathbf{W}^\top$  and the calibrated output  $\mathbf{X}_{\text{eval}} \hat{\mathbf{W}}_\lambda^\top$ . This metric provides a direct measure of the reconstruction error induced by the calibration process.

Figure 5a illustrates how the  $L_2$  error varies with  $\lambda$ , revealing a clear bias-variance trade-off. For small regularization values ( $\lambda < 10^{-3}$ ), the error is high and exhibits large variance across different calibration batches, indicating sensitivity to batch selection due to ill-conditioned activations. This suggests that when regularization is too weak, the inversion step amplifies noise, degrading output fidelity. On the opposite end, overly strong regularization ( $\lambda > 1$ ) significantly alters  $\hat{\mathbf{W}}_\lambda$ , reducing its ability to generalize and match the behavior of the original layer. In the intermediate range ( $10^{-3} \leq \lambda \leq 1$ ), the reconstruction error is minimized, indicating an optimal trade-off that stabilizes calibration without overfitting to the batch.

To further assess the impact of calibration on quantization, we analyze the quantization error of  $\hat{\mathbf{W}}_\lambda$ , as shown in

Figure 5b. The error is computed as

$$\frac{1}{d_{\text{out}} d_{\text{in}}} \|\hat{\mathbf{W}}_\lambda - \hat{\mathbf{W}}_{\lambda_Q}\|_F^2,$$

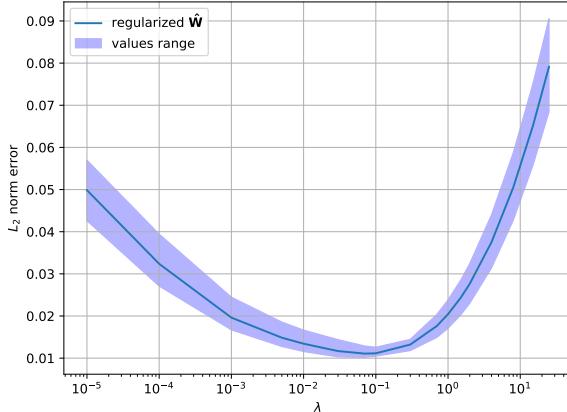
where  $\hat{\mathbf{W}}_{\lambda_Q}$  is the quantized version of the calibrated weight matrix using uniform quantization. The orange line in the figure represents the quantization error when using the original weight matrix  $\mathbf{W}$  instead of  $\hat{\mathbf{W}}_\lambda$ . We observe that counterintuitively for low  $\lambda$  values, the quantization error is extremely high due to the poor conditioning of  $\hat{\mathbf{W}}_\lambda$ , which leads to numerical instability and large weight magnitudes. As  $\lambda$  increases, particularly beyond  $5 \times 10^{-2}$ , the quantization error of the calibrated weights becomes consistently lower than that of the original weights. This suggests that regularization not only improves numerical stability, but also produces weight matrices that are better structured for quantization, exhibiting smaller dynamic ranges.

The selection of  $\lambda$  is not straightforward. One option is to choose the value that minimizes the reconstruction error in Figure 5a, which provides a good match to the original layer outputs. However, this choice does not necessarily lead to the lowest quantization error, and its impact on initialization quality for quantized training may be limited. Conversely, setting  $\lambda \approx 1$  improves the quantization behavior but can result in higher  $L_2$  output error, compromising the layer’s ability to generalize. This highlights the need to balance fidelity to the original outputs with improved robustness under quantization when selecting the regularization strength.

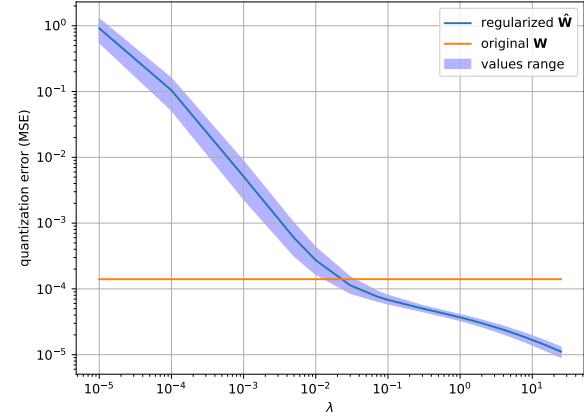
To select an appropriate value of  $\lambda$  in practice, we adopt a rule based on the singular value spectrum of the input matrix. If the activation matrix is ill-conditioned (i.e., with a condition number greater than  $10^2$ ) but not effectively rank deficient, we set  $\lambda$  to five times the smallest non-zero singular value. In the case of low-precision formats such as `bfloat16`, matrices with condition numbers exceeding  $10^3$  can already exhibit numerical instability. When the matrix is more severely ill-conditioned, with a condition number significantly greater than  $10^3$ , it is likely to be effectively rank deficient. In this case, we identify the smallest singular value  $\sigma_*$  such that  $\sigma_{\max}/\sigma_* \leq 10^3$ , and set  $\lambda = \lambda_0 \sigma_*$ .

As a general rule, depending on the conditioning characteristics of the specific network and input data,  $\lambda_0$  is chosen between two and five times the selected singular value. This strategy stabilizes the pseudoinverse computation while preserving the most meaningful directions in the activation space.

In Figure 6 we present the percentage reduction in standard deviation of all the layers after applying VRC to the B+ encoder when  $\lambda_0 = 2.0$ . In several blocks, some layers achieve a  $\sigma$  reduction exceeding 30%, which significantly lowers the initial quantization error. As detailed in Sec-



(a)  $L_2$  error over different  $\lambda$



(b) Quantization error (MSE) over different  $\lambda$

Figure 5. Calibration error for the linear layer `cut.blocks.21.mlp.layers.1` from the image encoder of the B+ model. In subfigure (a), we compare the output of the original linear layer with the output produced using the calibrated weights  $\hat{\mathbf{W}}_\lambda$ , measured via the  $L_2$  norm, across different values of the hyperparameter  $\lambda$ . In subfigure (b), we report the quantization error of  $\hat{\mathbf{W}}_\lambda$  for varying  $\lambda$ , relative to the quantization error of the original weights  $\mathbf{W}$ .

tion 3.2, the chosen  $\lambda_0$  value may be insufficient for certain layers; we observe that three layers specifically fallback to their original uncalibrated state.

## 6.2. Training and Other Results

### 6.2.1. Training Parameters

To ensure fairness, we used a consistent training and evaluation setup for both our Q-SAM2 and the baseline approaches Minmax [17], PACT [7], and T\_LSQ+ [5]. The training procedure follows the one proposed in the SAM2 [32] paper, where the authors referred to the "full training" step. The details are reported in Table 4 where we highlight the configurations and hyperparameters that are different from the original work. All trained models use the same image and video data, with a fixed random seed to ensure reproducibility.

Compared to the original SAM2 training setup, we introduce several modifications to accommodate hardware constraints and reduce training time. First, we reduce the maximum number of masks per frame from 64 to 60 to fit within GPU memory limits on an NVIDIA A100 80GB. We also skip the warm-up phase, reduce the amount of input data, and limit training to a single epoch.

All training experiments are conducted using 8 NVIDIA A100 80GB GPUs, with a total training time of less than 35 hours for the B+ architecture.

We adjust the main learning rate from  $4 \times 10^{-5}$  to  $1 \times 10^{-5}$  and decrease the effect of the gradient scaler by halving its initial scale value. For the T\_LSQ+ training, we reduce the scale and zero-point learning rate to  $5 \times 10^{-8}$  to stabilize parameter convergence.

The initial value for the LSC parameter  $k$  (Section 3.3) is

set to  $k = 2.5$  for B+ weights and activations, as well as for S and T activations. We use  $k = 2.8$  for the S and T weights. Finally, for observer calibration before QAT, we infer 300 images for Q-SAM2, PACT, and MinMax, but increase this to 900 images for T\_LSQ+ to capture its activation distributions more effectively.

### 6.2.2. Losses Plots

We adopt the standard loss of SAM2 [32] as  $\mathcal{L}_{\text{SAM2}}$ , which is composed of a linear combination of focal and dice losses for mask prediction, mean absolute error (MAE) loss for IoU prediction, and cross-entropy loss for object prediction. The weights are set in a fixed ratio of 20 : 1 : 1 : 1, respectively.

Figures 7 and 8 present the loss curves for the W2A4 and W2A2 quantization settings. Specifically, Figure 7 shows the results for the B+ model and one configuration of the S model (W2A2), while Figure 8 includes the second S configuration (W2A4) together with both configurations for the T model.

In all settings and across all loss components, Q-SAM2 demonstrates lower training losses compared to the MinMax, PACT, and T\_LSQ+ baselines. In addition, our model converges faster as the difference in loss values, particularly for the dice and IoU components, increases progressively with each training step.

### 6.2.3. Comparison with SAM2 training pipeline

The SAM2 is trained in two main stages followed by a fine-tuning procedure. The first stage, pre-training, involves training on static images using the SA-1B dataset. The second stage, called full training, uses a combination of images from SA-1B, videos from SA-V, and additional inter-

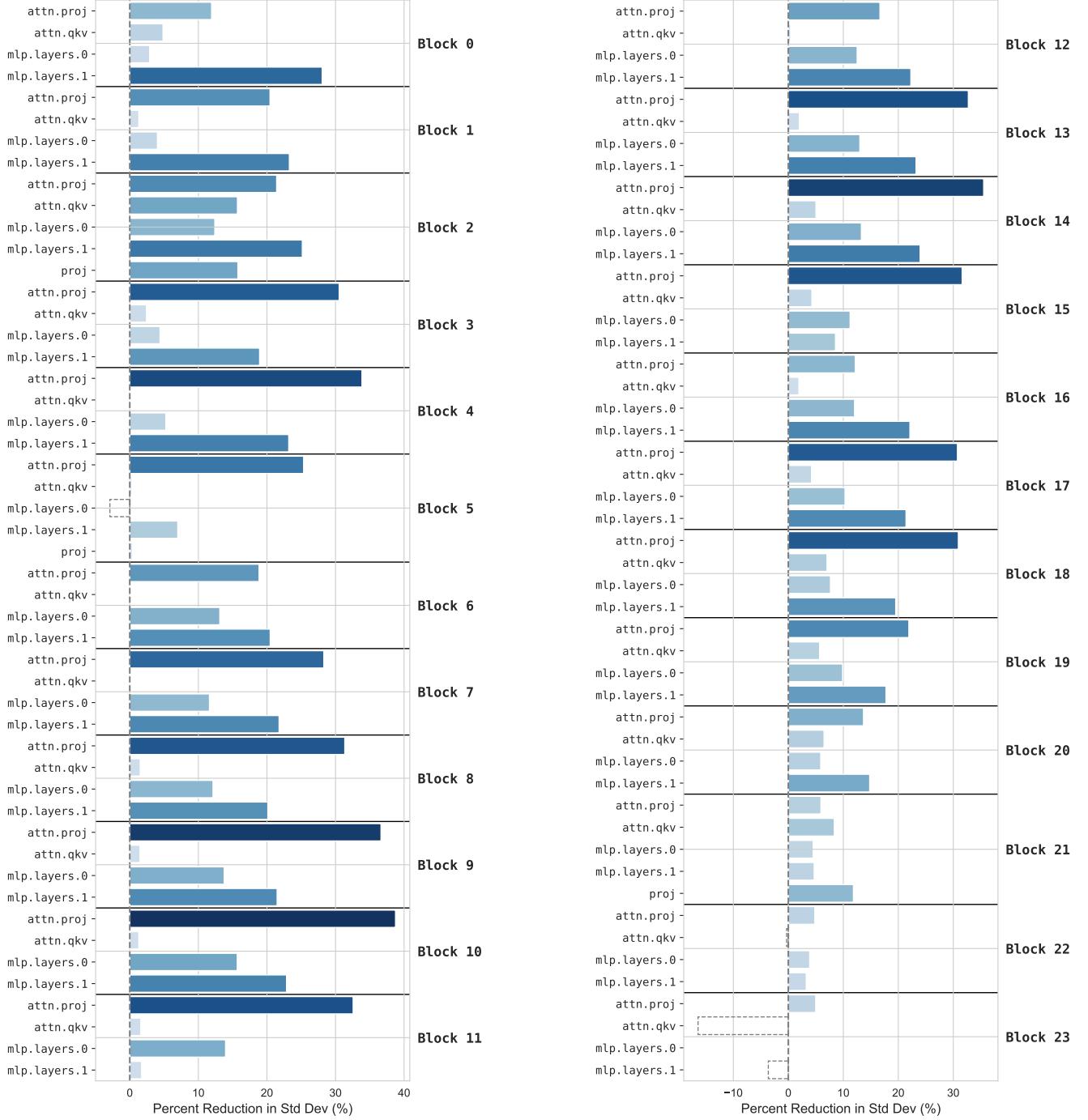


Figure 6. Impact of VRC on SAM2.1-B+ image encoder weight distributions for  $\lambda_0 = 2.0$ . VRC achieves a reduction up to 38.7%, with an average value of 13.2%.

nal data that is not publicly available. Finally, to improve segmentation performance on long video sequences, the authors fine-tune the model by increasing the number of input video sequences from 8 to 16. To accommodate this within GPU memory constraints, the image encoder is frozen dur-

ing this step. In our work, we reproduce the second stage, called "full training", focusing on a full architecture training with static image and video data while reducing both the data volume and the number of training steps with respect to the original paper. As demonstrated in Section 6.2.2, our

Configuration	Value
data	<b>SA-1B(8%) SA-V(45%)</b>
resolution	1024
precision	bfloat16
epochs	<b>1</b>
optimizer	AdamW
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
gradient	clipping type: $L_2$ , max: 0.1
weight decay	0.1
learning rate (lr)	<b>img. enc.: <math>1e - 5</math>, other: <math>3.0e - 4</math></b>
lr schedule	cosine
warmup	<b>no warmup</b>
layer-wise decay	0.8 (T, S), 0.9 (B+)
image augmentation	hflip, resize to 1024 (square)
video augmentation	hflip, affine (deg: 25, shear: 20), colorjitter (b: 0.1, c: 0.03, s: 0.03, h: null), grayscale (0.05), per frame colorjitter (b: 0.1, c: 0.05, s: 0.05, h: null), mosaic-2x2 (0.1)
batch size	256
drop path	0.1 (T, S), 0.2 (B+)
mask losses (weight)	focal (20), dice (1)
IoU loss (weight)	$L_1$ (1)
occlusion loss (weight)	cross-entropy (1)
max. masks per frame	<b>image: 60, video: 3</b>
# correction points	7
global attn. blocks	5-7-9 (T), 7-10-13 (S), 12-16-20 (B+)

Table 4. Hyperparameters and details of our Q-SAM2 and baselines QAT training for the three image encoder sizes B+,S,T.

method learns faster than baselines and achieves accurate results with fewer data, thus reducing training time. However, a direct comparison with the original SAM2 model remains difficult due to differences in training procedures. We plan to explore a fully aligned comparison with the complete SAM2 pipeline in future work.

#### 6.2.4. Comparison with LSQ

We compare Q-SAM2 with LSQ+ [5], a widely adopted and robust QAT algorithm known to learn quantization step sizes and zero point via gradient-based optimization. Although LSQ+ has shown strong results on various tasks, Table 5 shows that Q-SAM2 significantly outperforms on all video and image benchmarks in the W4A4 (per-tensor) setting, achieving gains that exceed 10 ppt J&F for VOS, and 7.7 ppt mIoU for image segmentation. For the per-channel configurations, LSQ+, and LSQ [10] fail to converge, consistent with the limitations previously reported in [20]; results are thus omitted. The training curves for LSQ+ are represented in Figure 9 and show that LSQ+ begins with a large performance gap compared to Q-SAM2; while the gap narrows slightly over time, LSQ+ still converges to significantly lower accuracy. This suggests that LSQ+ may require more training data or epochs to achieve competitive results,

Table 5. Evaluation of the semi-supervised VOS and instance segmentation tasks for the B+ model and W4A4 (per-tensor) configuration. Our solution is compared with the LSQ+ [5] QAT algorithm.

Metric	Dataset	Method	FP	W4TA4
J&F	SA-V Val [32]	LSQ+ [5] <b>Q-SAM2</b>	78.1	56.4 <b>65.2</b>
	SA-V Test [32]	LSQ+ [5] <b>Q-SAM2</b>	78.2	56.6 <b>66</b>
mAP	MOSE Val [9]	LSQ+ [5] <b>Q-SAM2</b>	73.7	56.6 <b>67.5</b>
	COCO2017 [21]	LSQ+ [5] <b>Q-SAM2</b>	49.6	41.1 <b>47.9</b>
mIoU	COCO2017 [21]	LSQ+ [5] <b>Q-SAM2</b>	59.1	44.9 <b>52.6</b>

reinforcing the efficiency and stability of the Q-SAM2 approach.

#### 6.2.5. Comparison with PTQ methods

We benchmark our QAT solution, Q-SAM2, against relevant PTQ quantization baselines. Our comparison in-

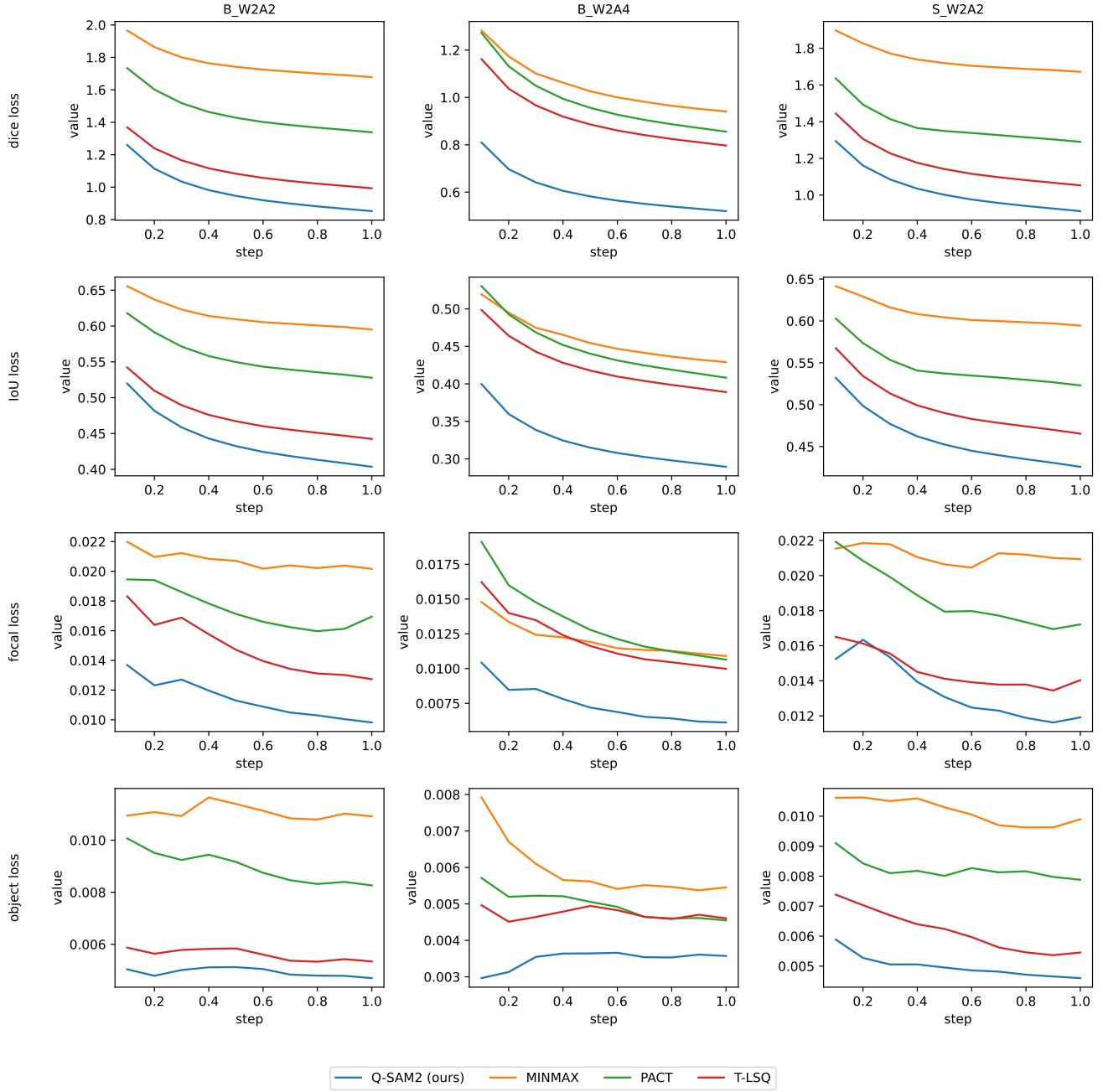


Figure 7. Training loss curves for image encoder size B+, S under two configurations W2A4 and W2A2. Each sub-plot contains a comparison between Q-SAM2 and the baselines for the four losses: dice, IoU, focal, and occlusion. Our solution outperforms the baselines in all losses for all configurations.

cludes PTQ4SAM [25], an algorithm specifically developed for the original SAM architecture [18]. We also report two leading general PTQ schemes, QDROP [34] and BRECCQ [19], known for their strong general performance. Table 6 summarizes the superior accuracy-compression trade-off achieved by Q-SAM2. We show

that our method pushes state-of-the-art performance into the ultra-low bit regime, offering significant size reductions without sacrificing fidelity. Q-SAM2 achieves exceptional compression, with the Q-SAM2-B+ model requiring only 38.8 MB for storing the weights. This results in a  $\sim 8.1\times$  reduction in model size compared to the largest baseline,

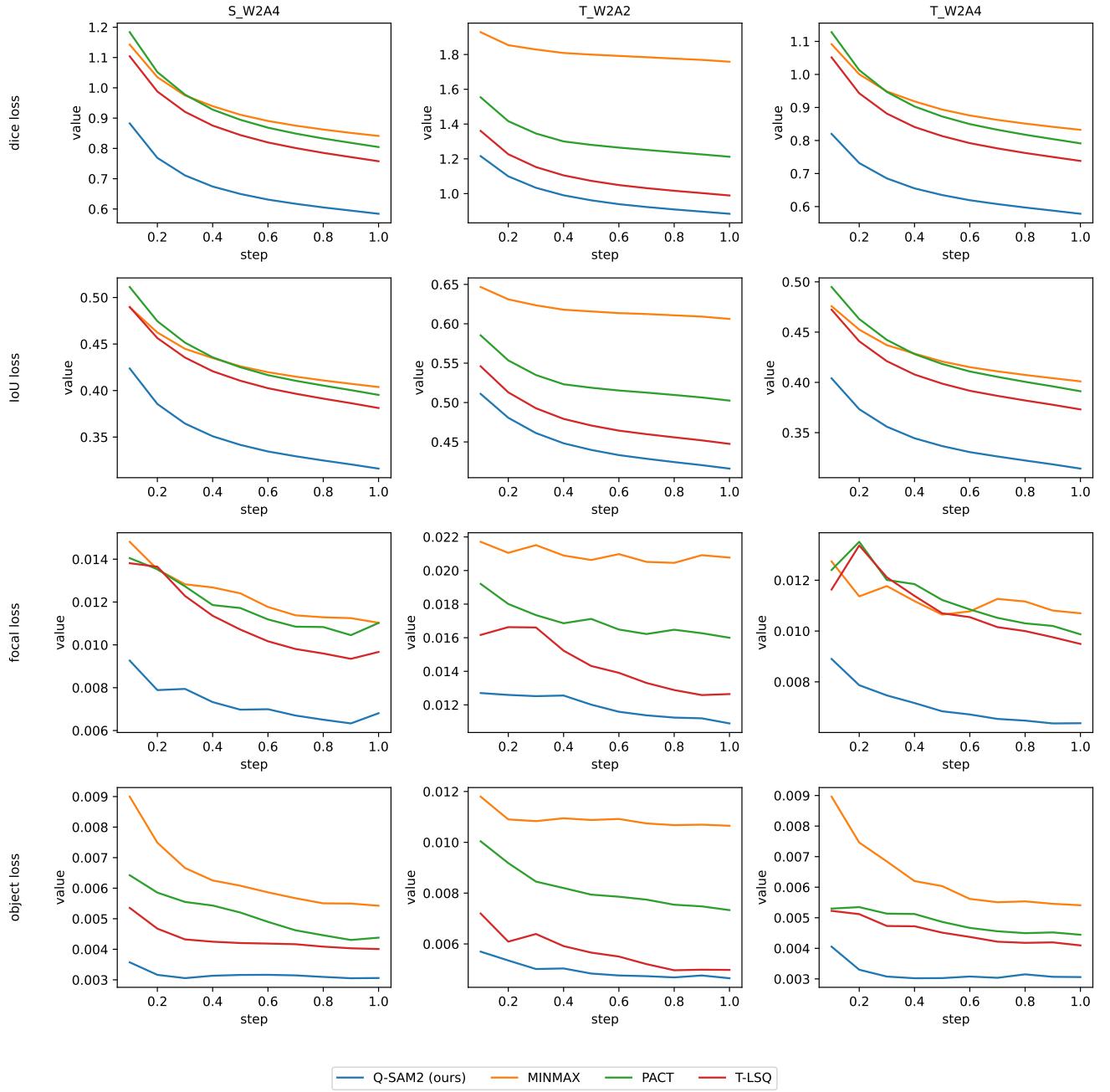


Figure 8. Continuation of the training loss curves from Figure 7 for image encoder size S, T under two configurations W2A4 and W2A2. Each sub-plot contains a comparison between Q-SAM2 and the baselines for the four losses: dice, IoU, focal, and occlusion. Our solution outperforms the baselines in all losses all configurations.

PTQ4SAM-H (W4A4, 317 MB). Our smallest Q-SAM2-S and Q-SAM2-T configurations further reduce size to approximately 30 MB. Crucially, the Q-SAM2 models retain accuracy far beyond similarly compressed or even larger baselines. The Q-SAM2-B+ (W2A4, 45.8 mAP) is competitive with the largest PTQ4SAM-H (W4A4, 43.9 mAP).

Furthermore, Q-SAM2-B+ remains robust even in the most aggressive W2A2 setting, retaining 39.2 mAP, which surpasses the W4A4 performance of the larger PTQ4SAM-L (W4A4, 36.6 mAP).

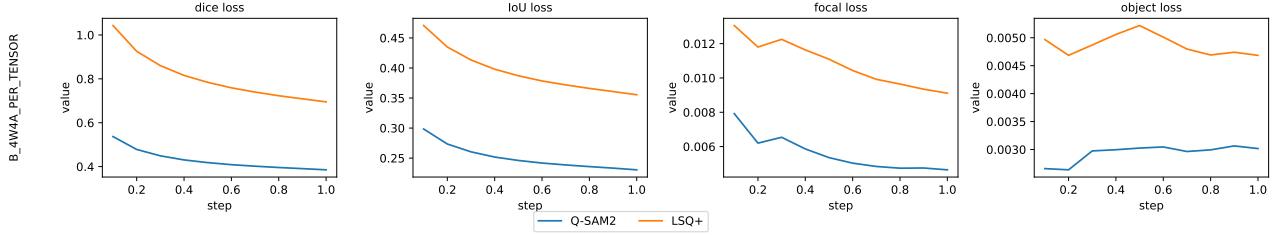


Figure 9. Training loss curves for image encoder size B+ under W4A4 (per-tensor) configurations. Each sub-plot contains a comparison between Q-SAM2 and the LSQ+ [5] method for the four losses: dice, IoU, focal, and occlusion.

Table 6. Comparison between Q-SAM2 and PTQ methods based on the original SAM [18].

Encoder	Precision	Size W (MB)	Method	mAP	FP
ViT-H	W6A6	477	PTQ4SAM [25] QDROP [34] BRECQ [19]	48.7 48.3 46.0	49.1
	W4A4	317	PTQ4SAM [25] QDROP [34] BRECQ [19]	43.9 41.7 17.6	
ViT-L	W6A6	230	PTQ4SAM [25] QDROP [34] BRECQ [19]	48.3 47.5 46.6	48.6
	W4A4	154	PTQ4SAM [25] QDROP [34] BRECQ [19]	36.6 27.5 12.3	
ViT-B	W6A6	71	PTQ4SAM [25] QDROP [34] BRECQ [19]	40.4 38.9 31.8	44.5
	W4A4	47	PTQ4SAM [25] QDROP [34] BRECQ [19]	14.4 11.2 3.6	
Hiera-B+	W2A4 W2A2	38.8 38.8	Q-SAM2	45.8 39.2	49.6
Hiera-S	W2A4 W2A2	32.0 32.0		44.3 36.1	48.4
Hiera-T	W2A4 W2A2	30.2 30.2		43.8 35.4	48.0

### 6.3. SAM2 Weight and Activation Distributions

As discussed in Section 3.1, the core challenge in quantizing SAM2 stems from extreme weight and activation distributions. This appendix provides the empirical foundation for that assertion and justifies the introduction of our VRC and LSC methods.

We quantify the severity of weight outliers in the B+ encoder by analyzing the scale difference between the weight distribution’s core and its extremes. Specifically, we examine the ratio of the standard deviation ( $\sigma$ ) to the absolute maximum value ( $\max(|W|)$ ). We found that on average the absolute maximum value is located at  $13.6\sigma$  with a peak of  $39.0\sigma$ . These results confirm that high-magnitude outliers reside far from the central mass of the distribution, consuming a disproportionate share of the quantization range.

Furthermore, we demonstrate that the quantization challenge extends significantly to activations. Beyond exhibit-

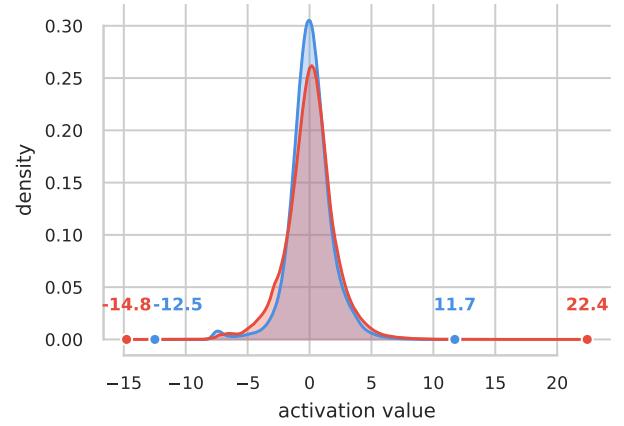


Figure 10. Comparison of the kernel density estimates across 50 random input generating the lowest (blue) and highest (red) total magnitude response in the *image\_encoder.trunk.blocks.1.mlp.layers.0* layer of the B+ model. Markers indicate the precise minimum and maximum values, highlighting the shift in dynamic range between distinct inputs.

ing large value ranges, we observe substantial distribution shifts within the same layer during inference. This inherent statistical instability, which we visualize in Figure 10, underscores the critical need for a stabilization mechanism like momentum-stabilized clipping. Building upon these

Table 7. Ablations of value  $\alpha$  from Equation 4 using model B+ and W2A4 configuration.

Dataset	Clipping ( $\alpha$ )	VRC	mIoU
COCO2017 [21]	✗	✗	39.0
	2.6	✗	43.3
	2.6	✓	44.7

observations, we investigate fixed clipping for weights and activations as an initial outlier mitigation strategy. This involves truncating values based on statistical bounds, specif-

ically using the rule

$$\mu \pm \alpha\sigma, \quad (4)$$

where  $\mu$  and  $\sigma$  are the distribution mean and standard deviation, and  $\alpha$  is a fixed scaling factor. We apply Equation 4 on a MinMax approach where we calculate the statistics per-tensor for activations and per-channel for weights, then applying a fixed  $\alpha = 2.6$ . Table 7 shows the results of this experiment, and confirms the effectiveness of statistical clipping, with a 4.3 ppt improvement in accuracy over the unclipped baseline. Furthermore, our VRC enhances the performance of the MinMax approach, demonstrating additional 1.4ppt gains beyond those achieved by simple static bounds.

#### 6.4. Qualitative Results

We have extensively benchmarked the original variants of SAM2, the MinMax baseline, and our Q-SAM2 algorithm. The variation includes different architectures, prompts, and quantization configurations for a fair comparison in real-world scenarios. Figure 11, Figure 12, and Figure 13 show the impact of adding more input points as prompt. We observe that our algorithm provides cleaner masks compared to PACT, and T\_LSQ+ algorithms with the same quantization scheme. We also note that by providing one prompt point only, typically all variants of the algorithms tend to provide a clean mask. However, and that is a well-studied behavior of the original SAM model, this mask might be only a part of the user’s intended selection, such as the front of the racing car as in Figure 12, or a single garlic bulb as in Figure 13. Henceforth, interesting examples include those that require multiple prompts to select the entire object. Those 3-point or 5-point prompts trigger differences among the quantized models. Compared against PACT, and T\_LSQ+, our 2-bit models require fewer prompt points to recover a larger fraction of the entire object, producing a cleaner and better result.

Figure 14, Figure 15 show the entire family of models including SAM2 base plus, small, and tiny architectures at full precision, as well as all variants of PACT, T\_LSQ+, and our Q-SAM2 models with 2-bit quantization for weights and 2 and 4 bits for activations. We first analyze the visual results prompted by a single point (Image 14). In this challenging scenario, many baseline models erroneously segment only the object’s head. Our Q-SAM2 family (W2A4 variants) and the B+W2A2 model yield segmentation maps comparable to the full-precision network, whereas other baselines struggle significantly. We examine the second image (prompted by three points), where nearly all models correctly identify the target object. However, a closer inspection reveals distinct qualitative differences between our proposed Q-SAM2 solution and the competing baselines.

Finally, Figures 16 and 17 present a qualitative comparison on a video from the SA-V test set for the semi-

supervised Visual Object Segmentation (VOS) task. The models included in this comparison are SAM2.1, the baselines, and our proposed Q-SAM2 for the W2A2 schema. The first column of Figure 16 shows the initial ground-truth mask used to prompt the models, as required in the semi-supervised VOS setup. Although T\_LSQ+ generally succeeds in segmenting the target object, Q-SAM2 produces cleaner masks with fewer border artifacts compared to the baselines.

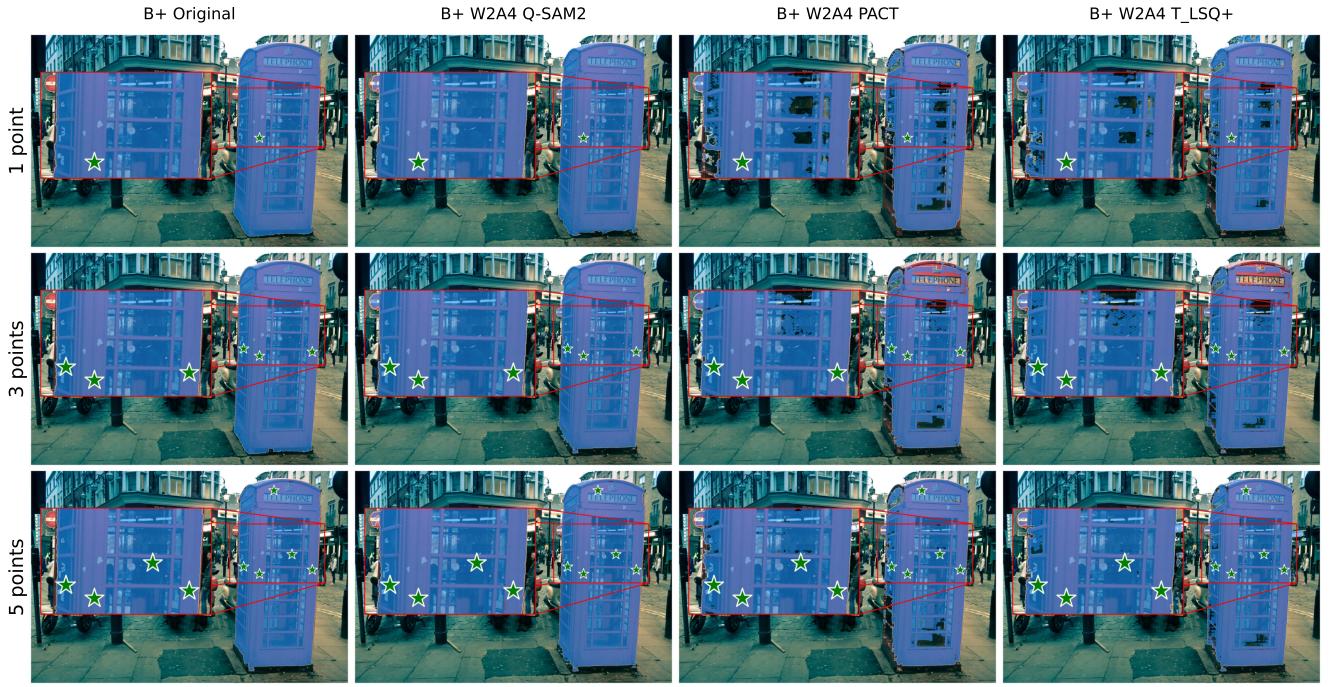


Figure 11. Qualitative results of SAM2, QSAM2(ours), PACT, and T\_LSQ+ on the base plus architecture. From top to bottom, we add 1, 3, and 5 prompt points.

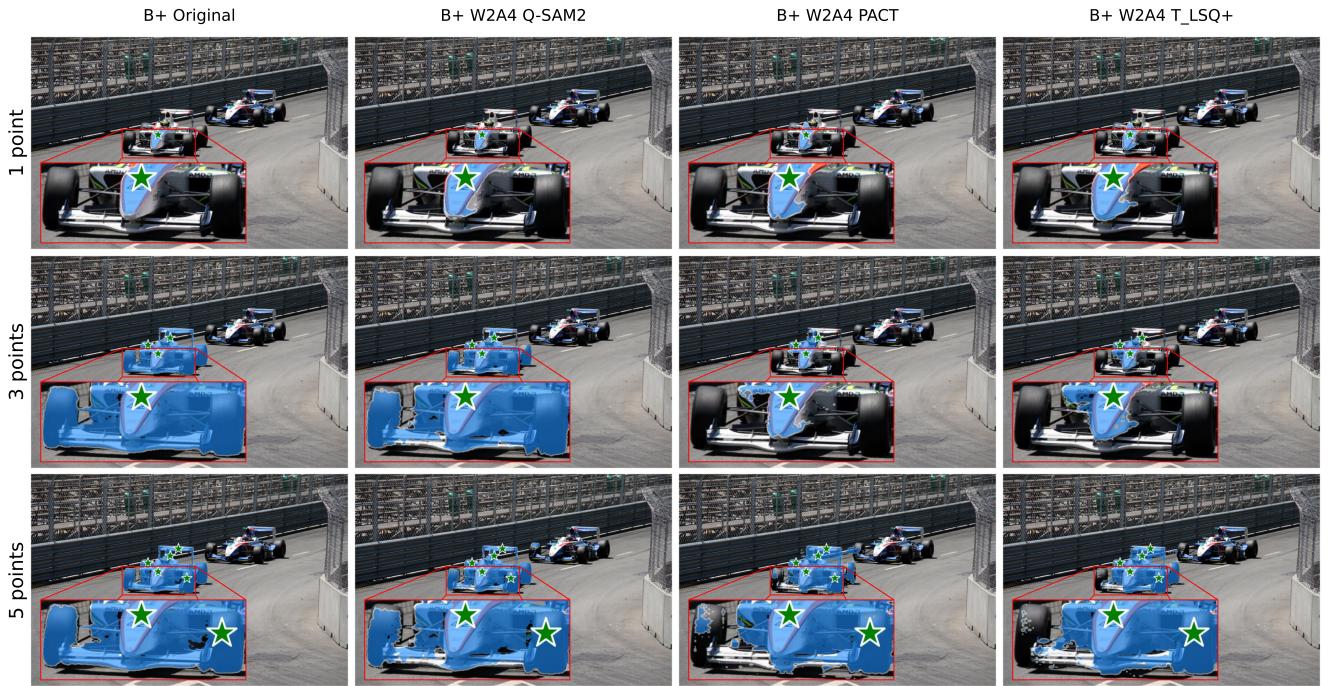


Figure 12. Qualitative results of SAM2, QSAM2(ours), PACT, and T\_LSQ+ on the base plus architecture. From top to bottom, we add 1, 3, and 5 prompt points.

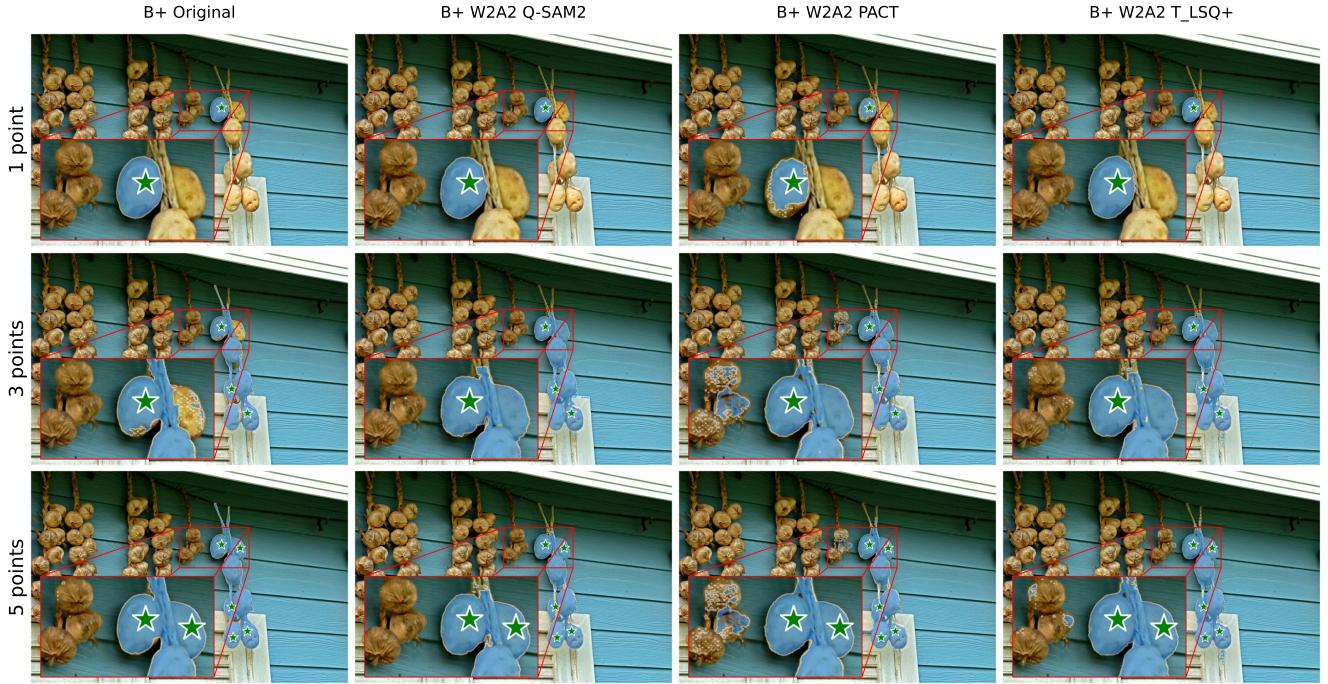


Figure 13. Qualitative results of SAM2, QSAM2(ours), PACT, and T\_LSQ+ on the base plus architecture. From top to bottom, we add 1, 3, and 5 prompt points.

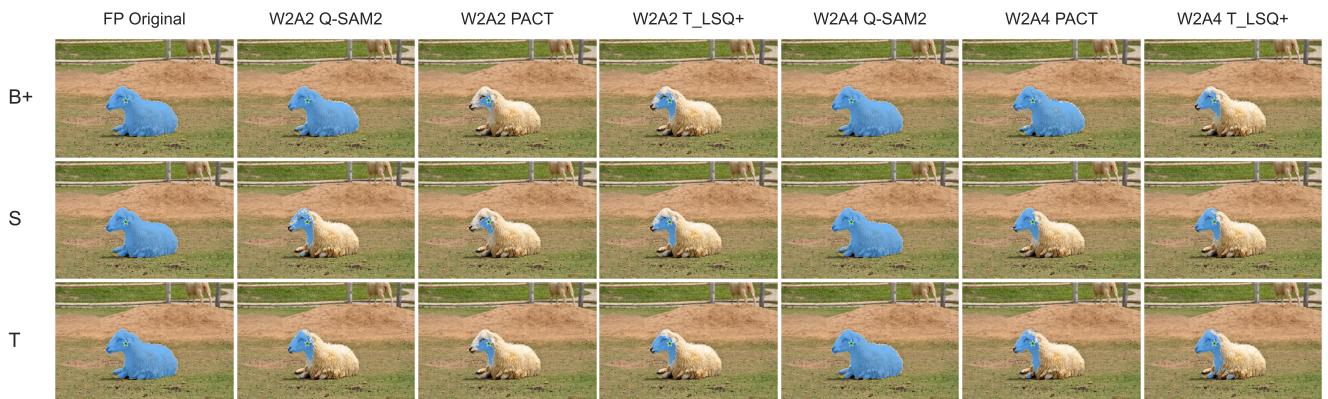


Figure 14. Qualitative results of variants of models with 1 point as prompt.

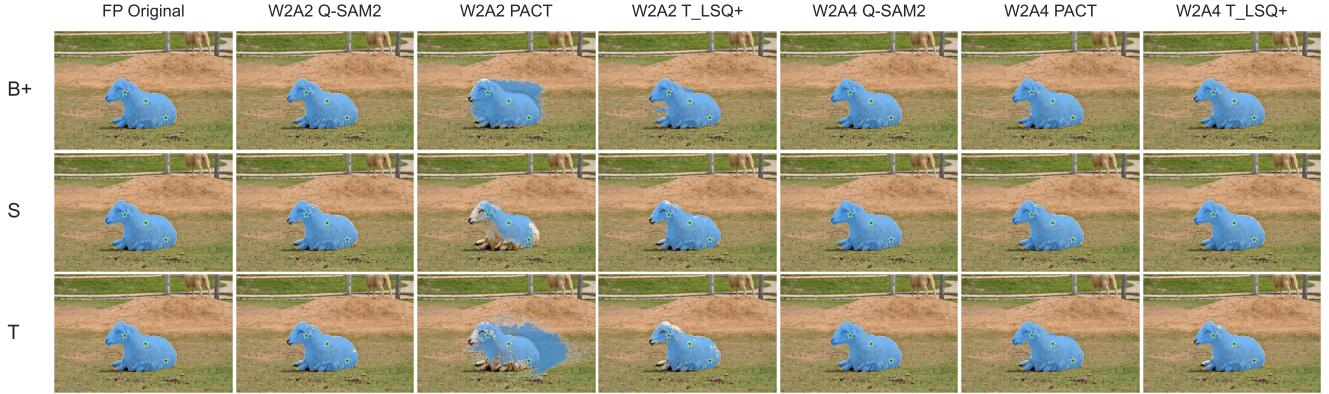


Figure 15. Qualitative results of variants of models with 3 points as prompt.

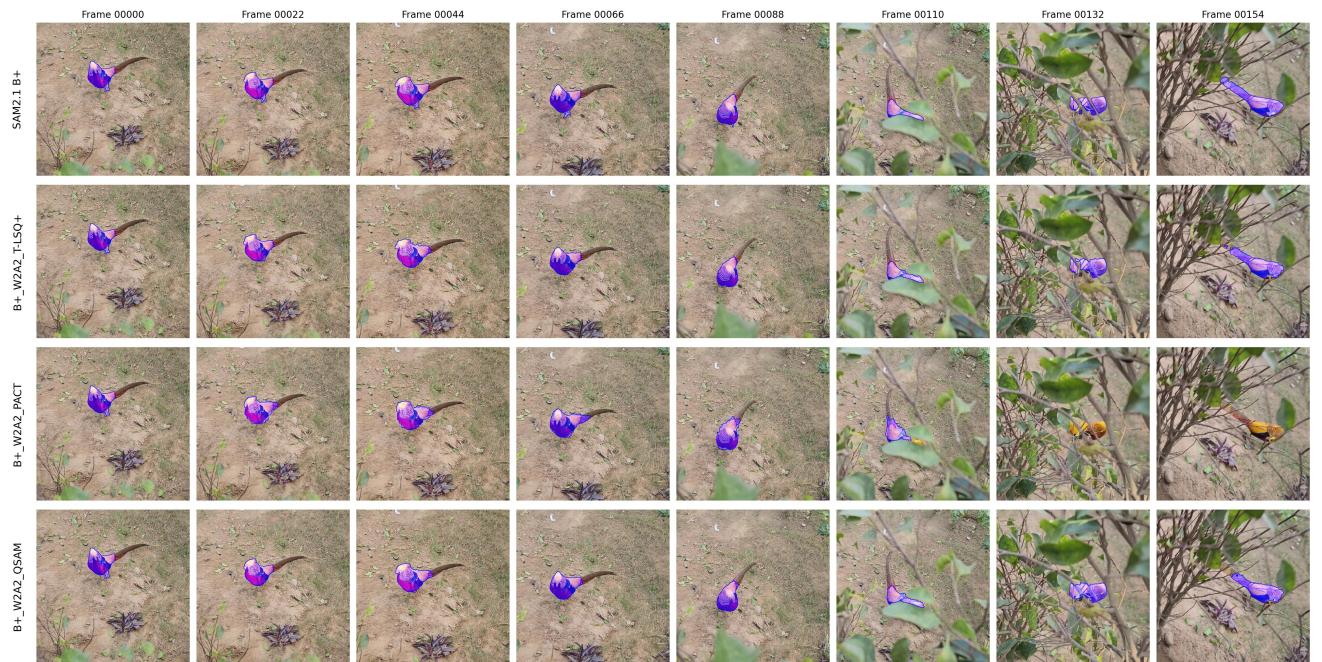


Figure 16. Qualitative results of variants of models SAM2, MinMax, and QSAM2(ours) on the base plus architecture for a video sequence. The first column shows the input mask provided in the first frame, while the remaining columns display the predicted segmentations for each model across subsequent frames.



Figure 17. Continuation of qualitative results from Figure 16, showing additional frames from the same video sequence.