SDA-SAM: Semantic-Driven Adaptive Mixed-Precision Quantization for Segment Anything Model

Wenbo Tan[1,2], Huimin Lu[1,2,*], Jianwei Guo[1,*], Zexing Zhang[3,*]
[1]Changchun University of Technology, China
[2]Jilin Province Science and Technology Innovation Center for Multimodal Cognitive Computing and Analysis of Medical Biometrics, China
[3]National University of Defense Technology, China
{2202403005@stu., luhuimin@, guojianwei@}ccut.edu.cn, zacheryzhang@163.com

***Abstract*—The Segment Anything Model (SAM) exhibits strong zero-shot segmentation capability, yet its parameter count and compute cost hinder deployment on edge devices. Post-training quantization (PTQ) is attractive for compressing SAM with only a small unlabeled calibration set, but existing PTQ pipelines typically adopt uniform bit-widths and are *semantic-blindness*: layers serving different semantic roles show markedly different quantization sensitivity, while reconstruction metrics (e.g., MSE) correlate poorly with task degradation. To address this, we propose SDA-SAM, a semantic-driven adaptive mixed-precision PTQ framework. Our key idea is to build a *Semantic Function Profile* (SFP) for each quantizable layer to guide bit allocation *before* quantization. SFP combines (i) a task-oriented *Gradient Sensitivity Score* computed from normalized gradient norms using pseudo-label self-training signals, and (ii) *Attention Entropy Analysis* that categorizes attention layers into localization- versus aggregation-dominant types. Guided by SFP, our Semantic-Driven Bit Allocation assigns higher precision to semantically critical layers and lower precision to robust layers under a global bit budget. On COCO instance segmentation with SAM-B, SDA-SAM matches the accuracy of uniform W6A6 while reducing the effective average weight bit-width to 5.1, and improves W4A4 accuracy by up to 8.5% compared to PTQ4SAM-S, without iterative reconstruction.**

***Index Terms*—Model quantization, Segment Anything Model, mixed-precision quantization**

I. Introduction

Segment Anything Model (SAM) [1] demonstrates strong zero-shot segmentation, but its memory and compute costs hinder edge deployment [2], [3]. Post-training Quantization (PTQ) requires minimal unlabeled data for compression, making it ideal for base model deployment [4], [5]. However, existing PTQ methods perform poorly on SAM. on COCO with YOLOX prompts, uniform W6A6 on SAM-B drops from 37.0% (FP) to 17.4% under a statistic-based PTQ baseline [6]. This paper investigates pre-quantization bit-width assignment per layer. Current uniform-bitwidth strategies [7] ignore semantic functional differences across SAM layers, which we identify as the key cause of performance degradation.

To deeply understand the *semantic-blindness* problem in SAM quantization, we conduct a systematic analysis of layer-level characteristics in SAM-B. As illustrated in Fig. 1:

Observation 1: Layers with different semantic functions exhibit significantly different quantization sensitivities. *Gradient sensitivity score* (GSS) is employed to quantify the contribution of each layer to the segmentation task. As shown in Fig. 1(a), the average GSS of the cross-attention layer in the mask decoder reaches 1.82, while the average GSS of the multi-layer Perceptron (MLP) layer in the image encoder is only 0.35, a difference of nearly 5-fold. The Pearson correlation between GSS and layer-wise quantization sensitivity reaches $r = 0.93$ (i.e., $r^2 = 0.87$), which greatly exceeds the correlation achieved by mean square error (MSE), where $r = 0.56$ (i.e., $r^2 = 0.31$). While MSE minimizes the reconstruction error, Fig. 1(d) shows that the GSS gradient-based task proxy achieves higher correlation in terms of task-oriented segmentation sensitivity compared to traditional reconstruction metrics. We define layer-wise quantization sensitivity as the performance drop $\Delta\mathrm{mAP}(l)$ when quantizing only layer l to W4A4 (others kept in FP16) on a fixed calibration/evaluation protocol.

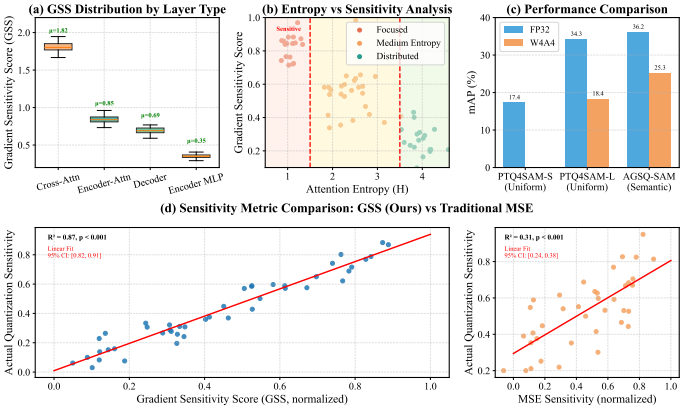

Fig. 1. Semantic Function Profile analysis. (a) Layer-wise GSS sensitivity differentiation; (b) Attention entropy vs semantic type mapping; (c) Accuracy-efficiency trade-off under mixed-precision bit allocation; (d) GSS vs MSE metric comparison.

Observation 2: Attention entropy effectively characterizes the functional semantic types of layers. As shown in Fig. 1(b), the entropy values clearly map to functional semantic types: the low entropy layer ($H < 1.5$) performs precise semantic localization with high quantization sensitivity; The high-entropy layer($H > 3.5$), performs context aggregation robustly. Unless stated otherwise, the thresholds (1.5, 3.5) are selected on a held-out calibration subset and remain stable across datasets; Section III-C further analyzes sensitivity to this choice.

Observation 3: Semantically guided mixed-precision bit allocation achieves a Pareto improvement of the precision-efficiency tradeoff. As shown in Fig. 1(c), the semantically guided scheme achieves 36.2% mAP under the same budget, which outperforms uniform 6-bit quantization (34.3%).

Based on the above analysis, we propose SDA-SAM, with the following main contributions:

- We identify *semantic-blindness* bit-width assignment as a key failure mode when applying PTQ to SAM, and propose a *Semantic Function Profile* (SFP) that characterizes each layer from both task-oriented sensitivity and attention-distribution type.
- We propose a *Semantic-Driven Bit Allocation* algorithm that assigns mixed precision under a global bit budget using SFP-ranked sensitivity, and is complementary to reconstruction-based PTQ.
- On COCO instance segmentation, SDA-SAM improves W4A4 accuracy by up to 8.5% over PTQ4SAM-S on SAM-B across multiple detectors, while requiring only a single forward–backward pass for profiling and significantly reducing calibration time compared to learning-based PTQ baselines.

Reproducibility. Upon acceptance, we will release code, configurations, evaluation scripts, and pretrained checkpoints to enable replication of our results.

II. Related Work

A. Segment Anything Model

SAM [1] demonstrates an excellent zero-shot generalization [8]. Its architecture consists of a Vision Transformer (ViT) [9] based image encoder, a lightweight cue encoder, and a bidirectional Transformer mask decoder. However, existing lightweight variants (e.g. EfficientSAM [10]) still take floating-point arguments and cannot take full advantage of integer arithmetic hardware acceleration.

B. Post-Training Quantization

Model quantization [4], [11] reduces storage and computational costs by converting floating-point values to low-order integers [12]. PTQ requires only a small amount of unlabeled data to calibrate the quantization parameter [13]. FQ-ViT [14] introduces a quadratic scaling factor to ViT to address the attention quantization challenge. NoisyQuant [15] alleviates the effect of activation outliers by noise bias. BRECQ [5] and QDrop [16] learn quantization parameters by minimizing the difference in response between the pre-quantization and post-quantization outputs. PTQ4SAM [6] solves the bimodal distribution of linear activation of the posterior key in the mask decoder. PTQ4SAM further reports a statistic-based variant (PTQ4SAM-S) and a learning-based variant (PTQ4SAM-L) built upon block-wise reconstruction with iterative optimization [6]. However, these methods all adopt a fixed bit-width strategy and ignore the layer-by-layer variation of quantization sensitivity.

C. Mixed-Precision Quantization

Mixed-precision Quantization (MPQ) [17] assigns different bit-widths to different layers. The HAWQ [18] family evaluates the layer sensitivity based on the Hessian matrix, while OMPQ [19] introduces a gradientguided strategy. DLSAQ [20] proposes layer adaptive quantization to alleviate the edge deployment problem. Mix-QSAM [21] explores the MPQ of SAM, but relies on refactoring optimization. Existing MPQ methods have limitations when applied to SAM: (1) the Hessian calculation brings huge overhead; (2) They are mainly designed for classification tasks, ignoring the unique semantic role of layers in segmentation tasks.

III. Method

This section elaborates on the SDA-SAM framework. Section III-A introduces preliminaries; Section III-B proposes the SFP method; Section III-C designs the SDBA algorithm and quantization execution details. The overall framework is shown in Fig. 2.

A. Preliminaries

Quantization Calibration. The quantization and dequantization processes of a uniform quantizer are defined as follows:

$$\text{Quant}: x_q = \text{clip}\left(\left\lfloor \frac{x}{s} \right\rceil + z, 0, 2^b - 1\right). \tag{1}$$

The quantization scale s and zero-point z are determined by clipping bounds x_{low} and x_{up}:

$$s = \frac{x_{up} - x_{low}}{2^b - 1}, \quad z = \left\lfloor -\frac{x_{low}}{s} \right\rceil, \tag{2}$$

where b denotes the bit-width. Existing methods adopt a uniform b value for all layers, neglecting the sensitivity differentiation caused by differences in semantic functions across layers. Specifically, given a pre-trained SAM model $\mathcal{M}$ containing L quantizable layers, an average bit budget $\bar{b}$, and a candidate bit set $\mathcal{B} = \{4, 5, 6, 7, 8\}$, the objective is to solve for the optimal bit allocation:

$$\max_{\{b_l\}_{l=1}^{L}} \mathcal{P}(\mathcal{M}_Q(\{b_l\})) \quad \text{s.t.} \quad \frac{\sum_{l=1}^{L} n_l \cdot b_l}{\sum_{l=1}^{L} n_l} \le \bar{b}, \quad b_l \in \mathcal{B}, \tag{3}$$

where $\mathcal{P}(\cdot)$ is the task performance metric and n_l is the parameter count of layer l. This paper directly characterizes the quantization properties of each layer through SFP, transforming the search problem into a greedy allocation problem based on sensitivity ranking.

B. Semantic Function Profiling

This section proposes the SFP framework, which characterizes layer-wise quantization properties from two complementary perspectives: task-oriented sensitivity and functional semantic type.

Task-Oriented Gradient Sensitivity Score. The gradient response of layer parameters to task loss directly reflects their contribution to the final performance. Since quantization inherently introduces bounded perturbations to the weights, gradient sensitivity can be used as an effective proxy for

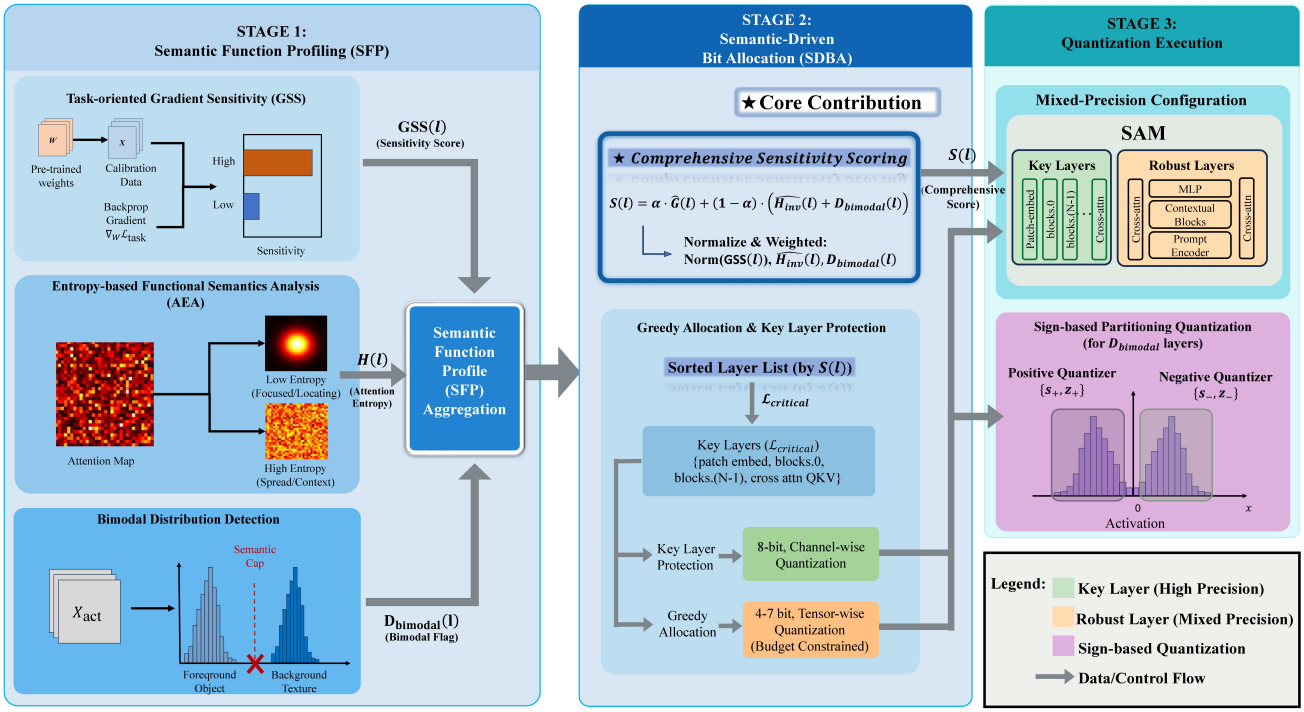

Fig. 2. Overall framework of SDA-SAM. The framework consists of three stages: (1) Semantic Function Profiling (SFP) construction through GSS and AEA; (2) Semantic-Driven Bit Allocation (SDBA) based on composite sensitivity scores; (3) Quantization execution with semantic critical layer protection.

quantization sensitivity. Since calibration data are unlabeled, we generate pseudo labels $\tilde{y}$ from the full-precision SAM and compute a self-training loss $\mathcal{L}(f(x,p),\tilde{y})$. We define GSS as:

$$\text{GSS}(l) = \mathbb{E}_{(x,p)\sim\mathcal{D}_{cal}}\left[\frac{\|\nabla_{W^{(l)}}\mathcal{L}(f(x,p),\tilde{y})\|_F}{\|W^{(l)}\|_F}\right], \quad (4)$$

where $\mathcal{L}$ is the segmentation loss and $\|\cdot\|_F$ is the Frobenius norm. Unlike spatially independent gradient measures [19], GSS incorporates SAM-specific spatial sampling mechanisms. Our design explicitly quantifies cue-driven sensitivity at the pixel level, thus capturing architecture-specific feature patterns critical for dense prediction tasks. Specifically, a uniform grid point cue p is generated for the input image x, the segmentation prediction is obtained through forward propagation, Using this prediction as a pseudo-label, the gradient of each layer is obtained after calculating the loss through backpropagation. The memory overhead is controllable by online updating the cumulative gradient statistics. Experiments show that there is a clear correspondence between the GSS of each layer of SAM-B and the type of semantic function: the cross-attention layer has the highest GSS (1.82), the deep encoder layer has the highest GSS (1.45), and the MLP layer has the lowest GSS (0.35). The essential defect of uniform quantization strategy is clearly exposed.

Entropy-Based Attention Analysis. The information entropy of the attention distribution directly reflects the type of semantic operation executed at a given layer. Let the attention inputs consist of the $X_Q \in \mathbb{R}^{N_q\times D}$ and $X_K, X_V \in \mathbb{R}^{N_k\times D}$ pairs, where N_q and N_k denote the number of tokens in the query and key, respectively, and D is the embedding dimension. The attention score matrix is computed as follows:

$$A_s = X_Q W_Q \cdot (X_K W_K)^T/\sqrt{d_h} \in \mathbb{R}^{N_q\times N_k}. \quad (5)$$

The attention scores are then normalized using the Softmax function to yield the attention weight matrix: $A_w \in \mathbb{R}^{N_q\times N_k}$, and the attention entropy is defined as:

$$H(A_w) = -\frac{1}{N_q}\sum_{i=1}^{N_q}\sum_{j=1}^{N_k} a_{i,j}\log(a_{i,j}+\epsilon), \quad (6)$$

where $a_{i,j}$ denotes attention weight elements and $\epsilon = 10^{-8}$ is a numerical stability term. This definition averages entropy across all query positions, characterizing the overall distributional properties of attention modules. For the l-th attention layer, the expected entropy is computed over the calibration set:

$$H(l) = \mathbb{E}_{(x,p)\sim\mathcal{D}_{cal}}\left[H(A_w^{(l)}(x,p))\right]. \quad (7)$$

Based on statistical analysis, we establish the mapping between entropy intervals and semantic function types: low-entropy layers ($H < 1.5$) perform semantic localization operations and are highly sensitive to quantization; high-entropy layers ($H > 3.5$) perform semantic aggregation operations and exhibit robustness. The robustness of high-entropy layers stems from weights approaching uniformity at each position ($a_{i,j} \approx 1/N_k$), where perturbations are statistically averaged, while perturbations at critical positions in low-entropy layers are directly amplified.

TABLE I

QUANTIZATION RESULTS FOR INSTANCE SEGMENTATION ON COCO. REC INDICATES WHETHER RECONSTRUCTION IS USED. WXAY DENOTES X-BIT WEIGHTS AND Y-BIT ACTIVATIONS. "-" DENOTES MAP BELOW 1%. BOLD AND UNDERLINED VALUES INDICATE THE BEST AND SECOND-BEST RESULTS UNDER THE SAME CONFIGURATION, RESPECTIVELY.

Detector	Method	REC	SAM-B			SAM-L			SAM-H		
			FP	W6A6	W4A4	FP	W6A6	W4A4	FP	W6A6	W4A4
Faster R-CNN	PTQ4SAM-S	×	33.4	15.4	-	36.4	35.7	18.1	37.2	36.0	24.1
	AdaRound	×		23.1	-		34.3	8.7		33.7	14.5
	QDrop	✓		29.3	13.0		35.2	22.6		36.3	32.3
	PTQ4SAM-L	✓		30.3	16.0		<u>35.8</u>	<u>28.7</u>		<u>36.5</u>	**33.5**
	AHCPTQ	×		<u>31.6</u>	11.7		35.6	27.4		36.3	31.4
	SDA-SAM (ours)	×		**32.4**	**23.1**		**35.9**	**30.2**		**36.6**	<u>32.3</u>
YOLOX	PTQ4SAM-S	×	37.0	17.4	-	40.4	40.0	20.6	41.0	40.3	26.7
	AdaRound	×		26.4	-		38.9	11.1		38.3	16.7
	QDrop	✓		33.6	13.3		39.7	25.3		40.4	35.8
	PTQ4SAM-L	✓		34.3	18.4		<u>40.3</u>	<u>31.6</u>		**40.7**	<u>37.6</u>
	AHCPTQ	×		<u>35.4</u>	13.4		40.0	31.0		40.4	35.2
	SDA-SAM (ours)	×		**36.2**	**26.9**		<u>40.2</u>	**32.1**		**40.7**	**37.9**
H-Deformable-DETR	PTQ4SAM-S	×	38.2	17.9	-	41.5	41.0	20.9	42.0	41.3	27.3
	AdaRound	×		27.2	-		39.9	8.0		39.4	16.3
	QDrop	✓		34.3	13.2		40.5	25.8		41.4	36.5
	PTQ4SAM-L	✓		35.1	17.3		**41.2**	32.1		<u>41.6</u>	**38.4**
	AHCPTQ	×		<u>36.6</u>	14.1		41.0	<u>32.3</u>		41.5	35.6
	SDA-SAM (ours)	×		**37.5**	**26.6**		**41.2**	**32.6**		**41.8**	<u>38.1</u>
DINO	PTQ4SAM-S	×	44.5	20.4	-	48.6	47.7	23.1	49.1	48.1	30.5
	AdaRound	×		31.2	1.2		46.6	8.8		46.0	18.2
	QDrop	✓		38.9	11.2		47.5	27.5		48.3	41.7
	PTQ4SAM-L	✓		40.4	14.4		<u>48.3</u>	36.6		<u>48.7</u>	**43.9**
	AHCPTQ	×		<u>41.9</u>	16.8		47.9	<u>36.6</u>		48.3	41.2
	SDA-SAM (ours)	×		**43.3**	**30.3**		**48.4**	**36.9**		**48.8**	<u>41.9</u>

Bimodal Distribution Detection and SFP Construction. The post-Key-Linear activations in SAM mask decoder exhibit pronounced bimodal distributions [7]. This work employs Kernel Density Estimation (KDE) to detect bimodality and applies sign-partition quantization to such layers. We define $D_{bimodal}(l) \in \{0, 1\}$ as an indicator returned by a KDE-based peak/valley test on post-Key-Linear activations (details in supplementary), where $D_{bimodal}(l) = 1$ triggers sign-partition quantization. Based on the above analysis, a complete SFP is constructed for each quantizable layer l:

$$\text{SFP}(l) = \{\text{GSS}(l), H(l), D_{bimodal}(l)\}, \tag{8}$$

SFP characterizes layer-wise quantization properties from two orthogonal dimensions: task importance and functional type. Compared to Hessian-based methods, SFP requires only a single forward-backward pass, significantly improving computational efficiency.

C. Semantic-Driven Bit Allocation and Quantization

Based on the SFP constructed in Section III-B, this section presents the SDBA algorithm to achieve layer-adaptive precision allocation under bit budget constraints.

Composite Sensitivity Score. The multi-dimensional semantic features are fused into a unified bit allocation criterion:

$$S(l) = \alpha \cdot \hat{G}(l) + (1 - \alpha) \cdot \left(\widehat{H_{inv}}(l) + D_{bimodal}(l) \right), \tag{9}$$

where each component is normalized as follows:

$$\hat{G}(l) = \frac{\text{GSS}(l) - \text{GSS}_{min}}{\text{GSS}_{max} - \text{GSS}_{min}} \in [0, 1], \tag{10}$$

$$\widehat{H_{inv}}(l) = \frac{H_{max} - H(l)}{H_{max} - H_{min}} \in [0, 1]. \tag{11}$$

The inverse entropy form $\widehat{H_{inv}}(l)$ ensures that low entropy (high sensitivity) maps to high scores, consistent with GSS semantics. Determined via coarse grid search on a calibration subset, these parameters yield stable performance across $\alpha \in [0.4, 0.6]$, Fig. 3 confirming robustness against overfitting rather than dependence on specific values.

Greedy Allocation Algorithm and Quantization Strategy. Given an average bit budget $\bar{b}$ and a candidate bit set $\mathcal{B}$, a greedy strategy assigns bit increments to high-sensitivity layers in descending order of sensitivity, with complexity of $O(L \log L + L \cdot \Delta b)$, completing in seconds for SAM-B [22]. To eliminate manual heuristics, we dynamically designate the top SFP-ranked layers as Semantically Critical Layers ($L_{critical}$). Statistical analysis confirms these layers align with patch embeddings, initial/final encoder blocks, and cross-attention QKV projections, consistent with the Information Bottleneck theory, removing protection causes 3.2% mAP degradation. Two protective measures are applied: (1) bit floor protection enforcing $bit \geq 8$. (2) per-channel quantization replacing per-tensor quantization. During quantization execution. Weight quantization employs asymmetric uniform quantization. We apply per-channel quantization to weights for all layers following common PTQ practice for transformers, and reserve additional protection to semantically critical layers via higher bit-width assignment. Activations uniformly adopt per-tensor asymmetric quantization. Bimodal layers employ sign-partition quantization. Following common practice, the first

TABLE II
SEMANTIC SEGMENTATION RESULTS ON ADE20K.

Model	Prec.	Method	REC	mIoU(%)
SAM-B	FP	-	-	33.17
	W6A6	PTQ4SAM-S	×	31.16
	W6A6	PTQ4SAM-L	✓	32.65
	W6A6	**SDA-SAM (ours)**	×	**32.78**
	W4A4	PTQ4SAM-S	×	31.08
	W4A4	PTQ4SAM-L	✓	31.85
	W4A4	**SDA-SAM (ours)**	×	**32.69**

and last layers retain FP16 precision to prevent accumulation of input-output precision loss [6].

IV. EXPERIMENTS

A. Experimental Setup

Tasks and Datasets. Experiments focus on instance segmentation and semantic segmentation. For instance segmentation, SAM generates segmentation masks based on bounding boxes provided by detectors, evaluated on Microsoft Common Objects in Context (MS-COCO) [23] using mean Average Precision (mAP). Semantic segmentation is evaluated on ADE20K [24] using mean Intersection over Union (mIoU).

Implementation Details. To ensure a fair comparison, we follow the standard experimental settings established in PTQ4SAM [6]. Specifically, we employ per-tensor asymmetric quantization for activations, per-channel asymmetric quantization for weights, keep the first and last layers unquantized, and use 32 calibration images.

Baseline Methods. Non-reconstruction-based: PTQ4SAM-S [6], AHCPTQ [25], and our SDA-SAM. Reconstruction-based: AdaRound [4], QDrop [16], and PTQ4SAM-L [6].

TABLE III
ABLATION STUDY ON SFP COMPONENTS (MAP%).

Configuration	W6A6	W4A4
Baseline	17.4	-
+GSS	33.8	22.1
+GSS+AEA	35.3	25.2
+GSS+AEA+Bimodal (Full SFP)	36.2	26.9

B. Instance Segmentation Results

We conduct detailed comparisons across different SAM scales (SAM-B/L/H) and various detectors (Faster R-CNN [26], YOLOX [27], H-Deformable-DETR [28], DINO [29]). Results on the COCO dataset are shown in Table I.

The proposed method demonstrates superior performance across all configurations. Compared to PTQ4SAM-S, SDA-SAM achieves nearly twofold mAP improvement on 6-bit SAM-B (DINO: from 20.4% to 43.3%) and recovers collapsed accuracy to usable levels at 4-bit. Compared to the reconstruction-based PTQ4SAM-L, our method achieves average improvements of 1.7% under W6A6 and 6.1% under W4A4, without requiring iteration. Consistent improvements on SAM-L and SAM-H further validate the generalization capability of the proposed method.

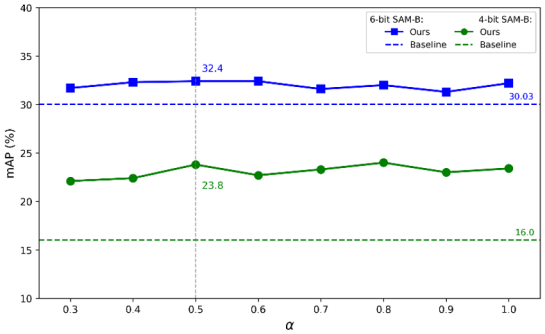

Fig. 3. Ablation study on composite sensitivity weight α.

C. Semantic Segmentation Results

For semantic segmentation, SegFormer is employed to provide category labels for SAM masks [30], with results shown in Table II. Our method achieves 32.78% mIoU at 6-bit and 32.69% mIoU at 4-bit, both outperforming the reconstruction-based PTQ4SAM-L.

D. Ablation Studies

Composite Sensitivity Weight Settings. In $S(l)$, we vary α from 0.3 to 1.0. Experiments show that $\alpha = 0.5$ is the optimal choice, and the method achieves satisfactory performance within the range of 0.4 to 0.6, demonstrating good robustness of this hyperparameter. Results are shown in Fig. 3.

TABLE IV
ENTROPY THRESHOLD SENSITIVITY ANALYSIS (W6A6 SAM-B MAP%).

Low-Entropy	High-Entropy	mAP(%)
1.0	3.0	35.8
1.5	3.0	35.9
1.5	**3.5**	**36.2**
1.5	4.0	36.0
2.0	4.0	35.6

Component Ablation and Efficiency Analysis. The performance of GSS alone, GSS combined with AEA, and the complete SFP is compared. The complete SFP achieves optimal performance under both precision settings, confirming the synergistic contribution of each component. Results are shown in Table III. We further verify that the gains of SDA-SAM persist when keeping the weight quantization granularity fixed (per-channel for all layers), indicating improvements mainly stem from semantic-driven bit allocation. Table IV presents the entropy threshold sensitivity analysis, demonstrating the method's robustness to threshold selection, with optimal performance at the (1.5, 3.5) configuration, in line with Gaussian Mixture Model (GMM) automatic clustering results.

Without requiring reconstruction or iterative optimization, SDA-SAM completes semantic profiling and quantization parameter calibration with a single forward-backward pass. The calibration process takes about 4 minutes (NVIDIA A100), compared to approximately 20k iterations and 45 minutes for PTQ4SAM-L—an order of magnitude improvement in efficiency, while maintaining superior quantization accuracy.

V. Conclusion

This paper proposes SDA-SAM, a semantic-guided dynamic post-training quantization framework for the Segment Anything Model. Aiming at the problem of *semantic-blindness* of existing methods, firstly, the SFP is constructed through the task-oriented gradient sensitivity component and the entropy-based attention analysis component to systematically represent the quantization attributes of each layer. Then, the SDBA algorithm implements layer adaptive bit configuration. The proposed method only needs one forward and backward traversal to complete the contour reconstruction, and the calibration efficiency is improved by about 10-fold compared with the method based on reconstruction. Experiments on MS-COCO and ADE20K datasets verify the effectiveness of the proposed method, and it has significant advantages especially in low-bit scenarios.Upon accepted publication of this paper, its source code/data will be made publicly available.

Limitations and Future Work. (1) GSS requires one backward pass; we plan to explore cheaper sensitivity proxies (e.g., diagonal Fisher or gradient-free perturbation estimates) to further reduce calibration cost. (2) Extending SFP to SAM2 and other foundation models may require revisiting entropy thresholds and bimodality tests due to architectural changes in tokenization and attention blocks.

Acknowledgment

This research was supported by Jilin Provincial Science and Technology Innovation Center for Multimodal Cognitive Computing and Analysis of Medical Biometrics (No. YDZJ202602CXJD038)

References

[1] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.

[2] Z. Chen, G. Fang, X. Ma, and X. Wang, "0.1% data makes segment anything slim," *CoRR*, 2023.

[3] Z. Zhang, H. Cai, and S. Han, "Efficientvit-sam: Accelerated segment anything model without performance loss," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7859–7863.

[4] M. Nagel, R. A. Amjad, M. Van Baalen, C. Louizos, and T. Blankevoort, "Up or down? adaptive rounding for post-training quantization," in *International conference on machine learning*. PMLR, 2020, pp. 7197–7206.

[5] Y. Li, R. Gong, X. Tan, Y. Yang, P. Hu, Q. Zhang, F. Yu, W. Wang, and S. Gu, "Brecq: Pushing the limit of post-training quantization by block reconstruction," *arXiv preprint arXiv:2102.05426*, 2021.

[6] C. Lv, H. Chen, J. Guo, Y. Ding, and X. Liu, "Ptq4sam: Post-training quantization for segment anything," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2024, pp. 15 941–15 951.

[7] J. Zhang, Z. Li, C. Hu, X. Liu, and Q. Gu, "Saq-sam: Semantically-aligned quantization for segment anything model," *arXiv preprint arXiv:2503.06515*, 2025.

[8] J. Ma, Y. He, F. Li, L. Han, C. You, and B. Wang, "Segment anything in medical images," *Nature Communications*, vol. 15, no. 1, p. 654, 2024.

[9] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM computing surveys (CSUR)*, vol. 54, no. 10s, pp. 1–41, 2022.

[10] Y. Xiong, B. Varadarajan, L. Wu, X. Xiang, F. Xiao, C. Zhu, X. Dai, D. Wang, F. Sun, F. Iandola *et al.*, "Efficientsam: Leveraged masked image pretraining for efficient segment anything," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16 111–16 121.

[11] Z. Li, X. Liu, B. Zhu, Z. Dong, Q. Gu, and K. Keutzer, "Qft: Quantized full-parameter tuning of llms with affordable resources," *arXiv preprint arXiv:2310.07147*, 2023.

[12] D. Jia, Y. Yuan, H. He, X. Wu, H. Yu, W. Lin, L. Sun, C. Zhang, and H. Hu, "Detrs with hybrid matching," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 19 702–19 712.

[13] J. Lin, J. Tang, H. Tang, S. Yang, W.-M. Chen, W.-C. Wang, G. Xiao, X. Dang, C. Gan, and S. Han, "Awq: Activation-aware weight quantization for on-device llm compression and acceleration," *Proceedings of machine learning and systems*, vol. 6, pp. 87–100, 2024.

[14] Y. Lin, T. Zhang, P. Sun, Z. Li, and S. Zhou, "Fq-vit: Post-training quantization for fully quantized vision transformer," *arXiv preprint arXiv:2111.13824*, 2021.

[15] Y. Liu, H. Yang, Z. Dong, K. Keutzer, L. Du, and S. Zhang, "Noisyquant: Noisy bias-enhanced post-training activation quantization for vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20 321–20 330.

[16] X. Wei, R. Gong, Y. Li, X. Liu, and F. Yu, "Qdrop: Randomly dropping quantization for extremely low-bit post-training quantization," *arXiv preprint arXiv:2203.05740*, 2022.

[17] R. Kumar *et al.*, "Hybrid and non-uniform quantization methods using retro synthesis data for efficient inference," *arXiv preprint arXiv:2012.13716*, 2020.

[18] Z. Dong, Z. Yao, A. Gholami, M. W. Mahoney, and K. Keutzer, "Hawq: Hessian aware quantization of neural networks with mixed-precision," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 293–302.

[19] Y. Ma, T. Jin, X. Zheng, Y. Wang, H. Li, Y. Wu, G. Jiang, W. Zhang, and R. Ji, "Ompq: Orthogonal mixed precision quantization," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 7, 2023, pp. 9029–9037.

[20] B. Zeng, B. Ji, X. Liu, J. Yu, S. Li, J. Ma, X. Li, S. Wang, X. Hong, and Y. Tang, "Lsaq: Layer-specific adaptive quantization for large language model deployment," *arXiv preprint arXiv:2412.18135*, 2024.

[21] N. Ranjan and A. Savakis, "Mix-qsam: Mixed-precision quantization of the segment anything model," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 3280–3290.

[22] Y. Li, H. Mao, R. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," in *European conference on computer vision*. Springer, 2022, pp. 280–296.

[23] T. Satoki, K. Hiroshi, I. Atsuki, S. Yasufumi, and Y. Fuyuka, "Greedy search algorithm for mixed precision in post-training quantization of convolutional neural network inspired by submodular optimization," in *Asian Conference on Machine Learning*. PMLR, 2021, pp. 886–901.

[24] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 633–641.

[25] W. Zhang, Y. Zhong, S. Ando, and K. Yoshioka, "Ahcptq: Accurate and hardware-compatible post-training quantization for segment anything model," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 22 383–22 392.

[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[27] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.

[28] D. Jia, Y. Yuan, H. He, X. Wu, H. Yu, W. Lin, L. Sun, C. Zhang, and H. Hu, "Detrs with hybrid matching," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 19 702–19 712.

[29] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," *arXiv preprint arXiv:2203.03605*, 2022.

[30] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in neural information processing systems*, vol. 34, pp. 12 077–12 090, 2021.