CAR-SAM: Cross-Attention Reconstruction for Post-Training Quantization of the Segment Anything Model

Supplementary Material

A. The Derivation of Matmul Compensation

A.1. Propagation to Q_{proj}.

To explicitly compensate for the quantization error introduced by the MatMul operation in attention computation, we formulate the following reconstruction objective. Let the quantized queries and keys be denoted as $\hat{Q}$ and $\hat{K}$, respectively. Our goal is to minimize the reconstruction discrepancy between the full-precision and quantized MatMul outputs:

$$\mathcal{L}^{\text{mse}} = \mathbb{E}\left[\|QK^\top - \hat{Q}\hat{K}^\top\|_2^2\right]. \tag{19}$$

To model the quantization-induced perturbations explicitly, we parameterize:

$$\hat{Q} = X_Q(W_Q + \delta W_Q), \quad \hat{K} = K + \delta K. \tag{20}$$

where X_Q is the full-precision input embedding, δW_Q denotes the compensation term, and $\delta K = \hat{K} - K$ captures the quantization error. Substituting these into Eq.19 yields:

$$\begin{aligned}\mathcal{L} = &\left\|QK^\top - X_Q(W_Q + \delta W_Q)(K + \delta K)^\top\right\|_2^2 \\ &+ \lambda\|\delta W_Q\|_2^2.\end{aligned} \tag{21}$$

where the last term introduces an L_2 regularization to prevent the compensation term δW_Q from overfitting or dominating the original weight W_Q. By expanding Eq.21 and computing the gradient of the reconstruction objective with respect to the compensation matrix δW_Q

$$\begin{aligned}&\frac{\partial \mathcal{L}}{\partial\, \delta W_Q} \\ &= 2\lambda\, \delta W_Q - 2X_Q^\top\left(QK^\top - X_Q\hat{W}_Q(K^\top + \delta K^\top)\right)(K + \delta K).\end{aligned} \tag{22}$$

Setting the derivative to zero yields the first–order optimality condition, and $\hat{K} = K + \delta K$:

$$2\lambda\, \delta W_Q - 2X_Q^\top\left(QK^\top - X_Q\hat{W}_Q\hat{K}^\top\right)\hat{K} = 0 \tag{23}$$

$$\Rightarrow \lambda\, \delta W_Q = X_Q^\top\left(QK^\top - X_Q\hat{W}_Q\hat{K}^\top\right)\hat{K} \tag{24}$$

Rearranging the above equality, we isolate the terms involving δW_Q:

$$\lambda\, \delta W_Q + X_Q^\top X_Q\, \delta W_Q(\hat{K}^\top\hat{K}) = X_Q^\top\left(QK^\top - X_Q W_Q\hat{K}^\top\right)\hat{K} \tag{25}$$

$$= X_Q^\top Q(K^\top - \hat{K}^\top)\hat{K} \tag{26}$$

$$= X_Q^\top Q\, \delta K^\top\hat{K} \tag{27}$$

Multiplying both sides by $(X_Q^\top X_Q)^{-1}$ from the left gives:

$$(X_Q^\top X_Q)^{-1}\lambda\, \delta W_Q + \delta W_Q(\hat{K}^\top\hat{K}) = (X_Q^\top X_Q)^{-1}X_Q^\top Q\, \delta K^\top\hat{K} \tag{28}$$

$$= W_Q\, \delta K^\top\hat{K}. \tag{29}$$

After rearranging the terms, we obtain a Sylvester-type matrix equation:

$$AX + XB = C. \tag{30}$$

where

$$\begin{aligned}&A = (X_Q^\top X_Q)^{-1}\lambda I, \quad B = \hat{K}^\top\hat{K}, \\ &C = W_Q\, \delta K^\top\hat{K}, \qquad X = \delta W_Q.\end{aligned} \tag{31}$$

A.2. Propagation to K_{proj}.

By taking similar steps, to propagate the quantization error of δQ into the key projection W_k, We rewrite the quantized key and query representations as

$$\hat{K} = X_K(W_K + \delta W_K), \quad \hat{Q} = Q + \delta Q. \tag{32}$$

Substituting these definitions into the reconstruction objective yields the following Eq. 19 :

$$\begin{aligned}\mathcal{L}^{\text{mse}} = &\left\|QK^\top - (Q + \delta Q)(W_K^\top + \delta W_K^\top)X_K^\top\right\|_2^2 \\ &+ \lambda\|\delta W_K\|_2^2.\end{aligned} \tag{33}$$

Taking the derivative with respect to δW_K yields:

$$2\lambda\, \delta W_K - 2X_K^\top\left(KQ^\top - X_K\hat{W}_K\hat{Q}^\top\right)\hat{Q} = 0 \tag{34}$$

$$\Rightarrow \lambda\, \delta W_K = X_K^\top\left(KQ^\top - X_K\hat{W}_K\hat{Q}^\top\right)\hat{Q} \tag{35}$$

Rearranging terms:

$$\begin{aligned}\Rightarrow &\lambda\, \delta W_K + X_K^\top X_K \delta W_K(\hat{Q}^\top\hat{Q}) \\ &= X_K^\top\left(KQ^\top - X_K W_K\hat{Q}^\top\right)\hat{Q} \\ &= X_K^\top K(Q^\top - \hat{Q}^\top)\hat{Q} \\ &= X_K^\top K\, \delta Q^\top\hat{Q}\end{aligned} \tag{36}$$

Finally, left-multiplying by $(X_K^\top X_K)^{-1}$ isolates δW_K:

$$\begin{aligned}\Rightarrow &(X_K^\top X_K)^{-1}\lambda I\, \delta W_K + \delta W_K(\hat{Q}^\top\hat{Q}) \\ &= (X_K^\top X_K)^{-1}X_K^\top K\, \delta Q^\top\hat{Q} \\ &= W_K\, \delta Q^\top\hat{Q}\end{aligned} \tag{37}$$

We get a same expression as 30 where:

$$A = (X_K^\top X_K)^{-1}\lambda I, \quad B = \hat{Q}^\top \hat{Q}, \\ C = W_K \delta Q^\top \hat{Q}, \qquad X = \delta W_K. \tag{38}$$

At this point, both the query- and key-side compensation equations share the same dependency on the inverse activation hessian matrices $(X_K^\top X_K)^{-1}$ and $(X_Q^\top X_Q)^{-1}$. We address this problem by performing singular value decomposition (SVD) on the matrix:

$$X^\top X = U \Sigma V^\top, \tag{39}$$

where $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$ contains the singular values in descending order. We select the smallest N such that the cumulative energy satisfies

$$\frac{\sum_{i=1}^{N} \sigma_i}{\sum_{i=1}^{n} \sigma_i} \geq t, \tag{40}$$

The ratio threshold t is empirically set to 0.1. and define λ as the mean of these top-N singular values:

$$\lambda = \frac{1}{N} \sum_{i=1}^{N} \sigma_i. \tag{41}$$

A.3. Propagation to V_{proj}.

To mitigate the degradation of the attention map A , the reconstruction objective is defined as:

$$\begin{aligned} &\mathcal{L} = \|AV - \hat{A}\hat{V}\|_2^2 \\ \Rightarrow\ &\mathcal{L} = \|AV - (A + \delta A) X_V (W_V + \delta W_V)\|_2^2 \\ &\quad + \lambda \|\delta W_V\|_2^2, \end{aligned} \tag{42}$$

where A denotes the full-precision attention map, $\hat{A}$ represents its dequantized counterpart, X_V is the quantized input to the value projection, W_V is the original value projection weight, and δW_V is the learnable compensation term. By taking the derivative of $\mathcal{L}$ with respect to δW_V and setting it to zero:

$$\begin{aligned} &\frac{\partial \mathcal{L}}{\partial\, \delta W_V} \\ &= 2\lambda\, \delta W_V - 2\hat{A} X_V (AV - \hat{A} X_V (W_V + \delta W_V)) \end{aligned} \tag{43}$$

$$\frac{\partial \mathcal{L}}{\partial\, \delta W_V} = 0 \tag{44}$$

$$\Rightarrow \lambda\, \delta W_V = X_V^\top \hat{A}^\top (AV - \hat{A} X_V (W_V + \delta W_V)) \tag{45}$$

$$\Rightarrow (X_V^\top \hat{A}^\top \hat{A} X_V + \lambda I)^{-1}\, \delta W_V = X_V^\top \hat{A}^\top (A - \hat{A}) V \tag{46}$$

we obtain the following closed-form solution:

$$\delta W_V = (X_V^\top \hat{A}^\top \hat{A} X_V + \lambda I)^{-1} X_V^\top \hat{A}^\top (A - \hat{A}) V, \tag{47}$$

B. The Proof of Joint Cross-Attention Reconstruction

B.1. The Derivation of Theorem 3.1

For each cross-attention module F_i, we minimize its corresponding MSE reconstruction loss defined as:

$$\mathcal{L} = \mathbb{E}\left[\left\| F_i(\hat{\mathbf{T}}, \hat{\mathbf{I}}, \hat{\mathbf{W}}) - F_i(\mathbf{T}, \mathbf{I}, \mathbf{W}) \right\|_2^F \right], \tag{48}$$

where $\mathbf{T}$ and $\mathbf{I}$ denote the prompt tokens and image embeddings. Formally, let the two consecutive cross-attention modules be denoted as f and g. The first token-to-image cross-attention module updates the prompt tokens $\mathbf{T}' = f(\mathbf{I}, \mathbf{T})$, and the subsequent image-to-token cross-attention refines the image embeddings $\mathbf{I}' = g(\mathbf{I}, \mathbf{T}')$. Let the ground-truth inputs be (T, I), and their quantized versions be $(T + \Delta T,\ I + \Delta I)$. The output perturbation is defined as

$$\Delta T' = f(T + \Delta T,\ I + \Delta I) - f(T, I). \tag{49}$$

Since ΔT and ΔI are small perturbations, a first-order Taylor expansion around (T, I) yields

$$f(T + \Delta T, I + \Delta I) \approx f(T, I) + J_f^{(T)} \Delta T + J_f^{(I)} \Delta I. \tag{50}$$

Hence the output error satisfies

$$\Delta T' \approx J_f^{(T)} \Delta T + J_f^{(I)} \Delta I \tag{51}$$

The subsequent image-to-token cross-attention takes (I, T') as input. Let the floating-point output be $I^* = g(I, T')$ and the quantized output be $\hat{I}' = g(I + \Delta I,\ T' + \Delta T')$. The induced quantization error is

$$\Delta I' = \hat{I}' - I^* = g(I + \Delta I, T' + \Delta T') - g(I, T'). \tag{52}$$

A first-order Taylor expansion gives

$$\Delta I' \approx J_g^{(I)} \Delta I + J_g^{(T')} \Delta T' \tag{53}$$

Substituting the expression of $\Delta T'$ from the first module into the second yields

$$\Delta I' = J_g^{(I)} \Delta I + J_g^{(T')} \left(J_f^{(T)} \Delta T + J_f^{(I)} \Delta I \right). \tag{54}$$

Collecting terms leads to the coupled first-order error propagation equation:

$$\Delta I' = \left(J_g^{(I)} + J_g^{(T')} J_f^{(I)} \right) \Delta I + \left(J_g^{(T')} J_f^{(T)} \right) \Delta T \tag{55}$$

And then we get Theorem 3.1:

$$\Delta I' \ \approx\ \underbrace{J_g^{(T')} \left(J_f^{(T)} \Delta T + J_f^{(I)} \Delta I \right)}_{\text{inter-branch-feedback term}} + J_g^{(I)} \Delta I \tag{56}$$

B.2. The Derivation of Corollary 3.2

The reconstruction loss in Eq. 48 is defined as

$$\mathcal{L} = \|\Delta I'\|_2^2. \tag{57}$$

This loss quantifies how quantization-induced deviations propagate across the two cross-attention modules. Among the two perturbation sources, only the upstream error $\delta f = \delta f(s_f)$ depends on the quantization scale s_f, while the downstream term δg reflects the intrinsic noise of the second module and therefore remains independent of s_f. This asymmetry is what enables a meaningful gradient signal for adjusting the upstream scale.

According the Throrem 3.1, the deviation at the output of the second cross-attention can be expressed as

$$\Delta I' \;\approx\; J_g^{(T')}\big(J_f^{(T)}\Delta T + J_f^{(I)}\Delta I\big) \;+\; J_g^{(I)}\Delta I, \tag{58}$$

For clarity, we introduce shorthand notations that group the two perturbation components: let δf collect the first-module contribution $J_f^{(T)}\Delta T + J_f^{(I)}\Delta I$, and let δg denote the second-module perturbation $J_g^{(I)}\Delta I$. Using these definitions, the expression can be rewritten as

$$\Delta I' = J_g^{(T')}\delta f + \delta g. \tag{59}$$

Using this expression, the loss becomes

$$\mathcal{L}(s_f) = \big(J_g^{(T')}\delta f + \delta g\big)^\top \big(J_g^{(T')}\delta f + \delta g\big), \tag{60}$$

which shows that the dependence on s_f enters exclusively through δf. Applying the standard identity for squared norms yields

$$\frac{\partial \mathcal{L}}{\partial s_f} = 2\,(\Delta I')^\top \frac{\partial \Delta I'}{\partial s_f}. \tag{61}$$

Since $J_g^{(T')}$ and δg do not depend on s_f, we have

$$\frac{\partial \Delta I'}{\partial s_f} = J_g^{(T')}\frac{\partial \delta f}{\partial s_f}. \tag{62}$$

Substituting this relation back into the gradient of the loss gives

$$\frac{\partial \mathcal{L}}{\partial s_f} = 2\big(J_g^{(T')}\delta f + \delta g\big)^\top J_g^{(T')}\frac{\partial \delta f}{\partial s_f}. \tag{63}$$

Because this quantity is scalar, the transpose can be moved to obtain the more compact form

$$\nabla_{s_f}\mathcal{L} = 2\,\big(J_g^{(T')}\big)^\top \big(J_g^{(T')}\delta f + \delta g\big)\frac{\partial \delta f}{\partial s_f}. \tag{64}$$

Variant	Setting	FP	BRECQ	QDrop	PTQ4SAM	**CAR-SAM**
SAM2-T	W6A6	74.6	70.3	72.1	73.6	**74.4**
	W4A4		52.8	62.8	67.8	**68.5**
SAM2-S	W6A6	74.6	71.4	71.7	73.1	**73.1**
	W4A4		43.9	62.2	67.3	**67.8**
SAM2-B+	W6A6	73.9	71.6	72.7	**73.1**	72.8
	W4A4		41.8	64.5	65.6	**66.6**
SAM2-L	W6A6	75.0	70.7	70.4	73.2	**73.4**
	W4A4		–	57.9	**65.6**	64.1

Table 6. Detection results of quantized SAM2 on COCO (vertical layout). FP denotes the full-precision reference. "–" indicates unavailable.

C. The Objection Detection Results of SAM2

Table 6 presents the object detection performance of various quantization methods applied to the SAM2 model family, evaluated on the COCO dataset. We report results across four SAM2 variants: SAM2-T, SAM2-S, SAM2-B+, and SAM2-L, under full-precision (FP), W6A6, and W4A4 settings.

In the 6-bit (W6A6) setting, all methods demonstrate acceptable retention of performance, with CAR-SAM consistently matching or slightly surpassing other baselines across most variants. For instance, CAR-SAM achieves 74.0 mAP on SAM2-T, comparable to the FP score of 74.6, and higher than BRECQ and QDrop in the same configuration. In the more challenging 4-bit (W4A4) regime, CAR-SAM maintains competitive accuracy across all variants and surpasses other methods on SAM2-B+ with a score of 66.6 and on SAM2-T with 68.0.

These results validate the robustness of our approach, particularly under aggressive quantization, and highlight its ability to preserve detection performance with minimal degradation compared to full precision.

D. Pseudocode of CAR-SAM

To help readers better understand our method, we provide pseudocode for solving the Sylvester equation as well as the overall pipeline.

Algorithm 1 Post-training quantization pipeline of CAR-SAM

Require: Full-precision SAM model $\mathbf{M}$, calibration set $\mathcal{D}_{cal}$
Ensure: Quantized model $\hat{\mathbf{M}}$

Initialize quantized model $\hat{\mathbf{M}}$ from $\mathbf{M}$
for each image-to-token cross-attention module $\mathbf{C}_i$ in $\hat{\mathbf{M}}$ **do**
  for $\mathbf{W} \in \{\mathbf{W}_Q, \mathbf{W}_K\}$ **do**
    Collect calibration activations from $\mathcal{D}_{cal}$
    Compute matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ ▷ see Eqs. (7) and (10)
    Solve the Sylvester equation $\mathbf{AX} + \mathbf{XB} = \mathbf{C}$ to obtain $\delta\mathbf{W}$ ▷ see Eq. (6)
    Update weights: $\mathbf{W} \leftarrow \mathbf{W} + \delta\mathbf{W}$
  end for
  for $\mathbf{W} = \mathbf{W}_V$ **do**
    Compute closed-form solution for $\delta\mathbf{W}$ ▷ see Eq. (12)
    Update weights: $\mathbf{W} \leftarrow \mathbf{W} + \delta\mathbf{W}$
  end for
end for
for each block $\mathbf{B}_i$ in $\hat{\mathbf{M}}$ **do**
  if $\mathbf{B}_i$ is a paired cross-attention block (f_i, g_i) **then**
    for each reconstruction step **do**
      Sample a mini-batch $\mathbf{X}_i$ from $\mathcal{D}_{cal}$
      Compute full-precision and quantized outputs of (f_i, g_i)
      Minimize the joint reconstruction loss:$\mathcal{L}_{rec} = \left\| F_{f_i,g_i}(\hat{T}, \hat{I}, \hat{W}) - F_{f_i,g_i}(T, I, W) \right\|_2^2$ ▷ see Theorem 3.2
      Update LSQ step sizes and quantization parameters jointly
    end for
  else
    for each reconstruction step **do**
      Sample a mini-batch $\mathbf{X}_i$ from $\mathcal{D}_{cal}$
      Compute full-precision and quantized outputs of $\mathbf{B}_i$
      Minimize the block-wise reconstruction loss: $\left\| F(\hat{\mathbf{x}}, \hat{\mathbf{w}}) - F(\mathbf{x}, \mathbf{w}) \right\|_2^2$
      Update LSQ step sizes and quantization parameters
    end for
  end if
end for
return $\hat{\mathbf{M}}$

```python
from scipy.linalg import solve_sylvester
def solve_Sylvester_equation(X_q, W_q, K, K_hat, lam=1e3, bias=None):
    """
    Solve the Sylvester equation for Q-branch compensation.

    The K-branch is handled in the same way by swapping Q and K,
    resulting in a corresponding Sylvester equation to solve for
    the compensation matrix of W_K.

    Args:
        X_q: input activation of Q projection, shape [..., d_in]
        W_q: Q projection weight, shape [d_out, d_in]
        K: full-precision key features, shape [..., d_out]
        K_hat: quantized/dequantized key features, shape [..., d_out]
        lam: regularization coefficient
        bias: optional bias of Q projection, shape [d_out]

    Returns:
        delta_W: compensation matrix for W_q
    """
    # flatten token dimensions
    X_q = X_q.reshape(-1, X_q.shape[-1])
    K = K.reshape(-1, K.shape[-1])
    K_hat = K_hat.reshape(-1, K_hat.shape[-1])

    # absorb bias into an augmented weight matrix
    if bias is not None:
        W_q = torch.cat([W_q, bias.unsqueeze(1)], dim=1).T
        X_q = torch.cat(
            [X_q, torch.ones(X_q.shape[0], 1, device=X_q.device, dtype=X_q.dtype)],
            dim=1
        )
    else:
        W_q = W_q.T

    # construct Sylvester equation: A X + X B = C
    A = lam * (X_q.T @ X_q)
    B = K_hat.T @ K_hat
    C = W_q @ ((K - K_hat).T @ K_hat)

    # solve on CPU with scipy
    delta_W = solve_sylvester(
        A.detach().cpu().numpy(),
        B.detach().cpu().numpy(),
        C.detach().cpu().numpy()
    )

    return torch.from_numpy(delta_W).to(X_q.device).to(X_q.dtype)
```

Figure 7. Pytorch-like code snippet of the Q-branch MatMul-aware compensation. The function constructs the Sylvester equation $AX + XB = C$ and solves for the compensation matrix ΔW.