Semantic Alignment and Reinforcement for Data-Free Quantization of Vision Transformers

Supplementary Material

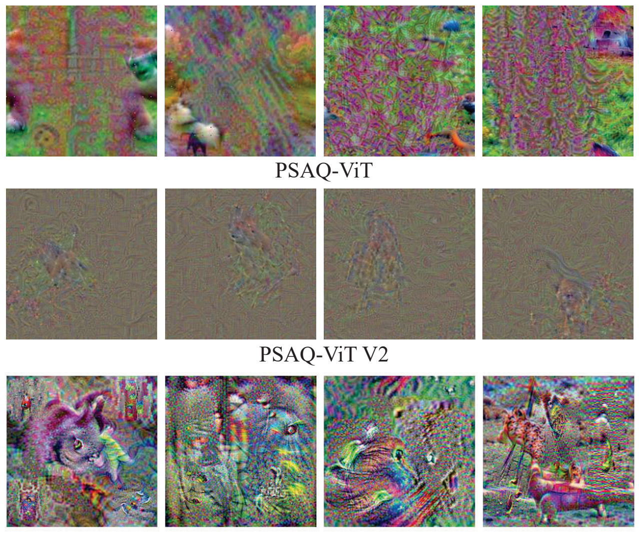

Figure 6. More synthetic images from PSAQ-ViT, PSAQ-ViT V2, and our SARDFQ.

```python

from scipy.stats import multivariate_normal
import numpy as np

def Single_Gaussian():
    mean = np.random.uniform(-5, 5, 2)
    covariance = None
    while True:
        A = np.random.uniform(0, 1, (2, 2)) * 10
        covariance = np.dot(A, A.T)
        try:
            eigenvalues, eigenvectors = np.linalg.eig(covariance)
            eigenvalues = np.clip(eigenvalues, 0, 5)
            covariance = np.dot(np.dot(eigenvectors,
                      np.diag(eigenvalues)), eigenvectors.T)
            return multivariate_normal(mean, covariance)
        except np.linalg.LinAlgError:
            continue
```

A. Code to Gaussian Distribution Geneartion

In the following code, we provide the Python code for generating the Gaussian used in our paper.

B. More Synthetic Images Comparisons.

Fig.,6 shows synthetic images (224×224) from PSAQ-ViT, PSAQ-ViT V2, and our SARDFQ. In comparison, images generated by PSAQ-ViT and PSAQ-ViT V2 contain many dull regions, such as the central parts of PSAQ-ViT and most of PSAQ-ViT V2. These dull areas typically lack content diversity, with simplified and repetitive textures, indicating semantic inadequacy [33]. In contrast, SARDFQ images exhibit greater content diversity and more complex textures, suggesting adequate semantics.

We also emphasize that although the generated images often have low sensory quality due to the presence of hard-to-identify, unreal textures [77], they still perform well in downstream tasks. As demonstrated in [13, 14, 84], recent zero-shot quantization methods (TexQ [13], Qimera [14], Intraq [84], etc.) produce images with low sensory quality, yet these methods achieve substantial performance improvements. Therefore, the effectiveness of generated images for downstream tasks should not be solely judged by their sensory quality. Thus, we encourage readers to focus on the semantic distribution extracted by the model, as illustrated in Fig. 1a of the main paper.

C. More Ablation Study

The ϵ_1 and ϵ_2 in SL balance the smoothness of the proposed SL loss. Tab. 6a demonstrates that the optimal performance is achieved when $\epsilon_1 = 5$. Incrementally increasing ϵ_1 from 1 to 5 improves performance from 61.00% to 62.29%. Then, further increasing ϵ_1 subsequently degrades performance. For example, increasing ϵ_1 to 9 results in 0.93% degradation. Tab. 6b presents the ablation study of ϵ_2. It can be seen that the optimal performance is achieved when $\epsilon_2 = 10$. Using smaller or larger values than 10 will hurt the performance.

D. Additional Quantization Results

D.1. Classification Results

Tab. 5 presents the quantization results for the W8A8 and W4A8 settings. Here, we use a linear quantizer for attention scores, which is consistent with the compared methods. The results of the compared methods are copied from their paper. It can be observed that the proposed SARDFQ outperforms the compared methods in most cases, except for W8A8 DeiT-T, where SARDFQ exhibits only a 0.05% gap. Notably, SARDFQ generally provides larger performance gains in the W4A8 setting. For instance, SARDFQ achieves increases of 1.24% and 0.70% on W4A8 DeiT-S and DeiT-B, respectively. These results clearly demonstrate the effectiveness of the proposed SARDFQ.

D.2. Detection and Segmentation Results

In Tab. 7, we provide the results on detection and segmentation tasks. Note that results with "†" are re-produced by ours and the other results of PSAQ-ViT V2 [48] and CLAMP-ViT [62] are copied from their paper. Following

Model	W/A	PSAQ-ViT [47]	PSAQ-ViT V2 [48]	SMI [33]	CLAMP-ViT [62]	SARDFQ (Ours)
ViT-S	8/8	31.45	-	-	-	**80.81**
(81.39)	4/8	20.84	-	-	-	**78.49**
ViT-B	8/8	37.36	-	-	84.19	**84.22**
(84.54)	4/8	25.34	-	-	78.73	**82.19**
DeiT-T	8/8	71.56	72.17	$70.27_{70.13}$	72.17	72.12
(72.21)	4/8	65.57	68.61	$64.28_{64.04}$	69.93	**70.05**
DeiT-S	8/8	76.92	79.56	-	79.55	**79.81**
(79.85)	4/8	73.23	76.36	-	77.03	**78.27**
DeiT-B	8/8	79.10	81.52	$75.99_{77.51}$	-	**81.83**
(81.85)	4/8	77.05	79.49	$78.58_{79.63}$	-	**80.19**
Swin-T	8/8	75.35	80.21	-	81.17	**81.18**
(81.35)	4/8	71.79	76.28	-	80.28	**80.34**
Swin-S	8/8	76.64	82.13	-	82.57	**82.85**
(83.20)	4/8	75.14	78.86	-	**82.51**	**82.51**

Table 5. Quantization results on ImageNet dataset, with top-1 accuracy (%) reported. The performance of the full-precision model is listed below the model name, "W/A" denotes the bit-width of weights/activations. The results of PSAQ-ViT [47], PSAQ-ViT V2 [48], SMI [33], and CLAMP-ViT [62] are copied from their paper, and "-" denotes the results are unavailable. For SMI [33], we provide the performance of using dense (normal-sized numbers) and sparse (smaller-sized numbers) synthetic images, respectively.

ϵ_1	Top-1
1	61.00
3	61.05
5	**62.29**
7	61.15
9	61.36

(a)

ϵ_2	Top-1
6	61.62
8	61.88
10	**62.29**
12	62.13
14	61.10

(b)

Table 6. Effect of varying (a) ϵ_1; (b) ϵ_2.

[48, 62], the results are evaluated on Mask R-CNN and Cascade R-CNN models, both with Swin-S as the backbone. The proposed SARDFQ consistently outperforms the compared methods. For example, SARDFQ achieves 1.2 and 2.9 increases on box and mask AP for W4A8 Mask R-CNN, respectively. For W4A8 Cascade R-CNN, SARDFQ achieves 0.5 and 0.2 increases on box and mask AP for W4A8 Mask R-CNN, respectively.

E. Practical Efficiency

In this subsection, we employ ViT-B as the example to provide practical efficiency results. For SARDFQ, if quantizing ViT-B using 3090 GPU, the data synthesis stage consumes 11887 MB GPU memory and 13 minutes, while the quantized network learning consumes 3324 MB GPU memory and 14 minutes.

For the quantized ViT-B, inference efficiency is only related to a specific bit-width. Tab. 8 provides efficiency metrics by evaluating the 4-bit ViT-B on 3090 GPU. We implement the 4-bit CUDA kernel by using *Cutlass library* and run the model on 3090 GPU with 200 images. For the FP model, the model size is 346 MB, the inference speed is 534 ms, and the running memory is 1859 MB. For the 4-bit model, the model size is 43.3 MB, the inference speed is 244 ms, and the running memory is 1165 MB. These results highlight the practical benefits of 4-bit quantization in terms of storage and computational efficiency.

F. Other Clarification

F.1. Why using Cosine similarity

We chose cosine similarity since it is a widely used and intuitive metric to measure feature alignment [21, 23, 66]. We also emphasize that cosine similarity ranges from -1 to 1, making values like 0.32 and 0.41 in Tab. 1 of the main paper reasonable.

F.2. Visualization of attention priors and generated images

Fig. 7 show synthetic images and their corresponding attention priors (the most salient generated map used in the last attention layer), demonstrating effective enhancement of attended regions.

Model	**W/A**	**Method**	**AP (box)**	**AP (mask)**
Mask R-CNN (Swin-S)	32/32	-	48.5	43.3
	4/8	PSAQ-ViT V2† [48]	44.7	39.0
		SARDFQ (Ours)	**45.9**	**41.9**
	8/8	PSAQ-ViT V2† [48]	47.8	42.7
		SARDFQ (Ours)	**48.2**	**43.0**
Cascade R-CNN (Swin-S)	32/32	-	51.8	44.7
	4/8	PSAQ-ViT V2 [48]	47.9	41.4
		PSAQ-ViT V2† [48]	48.2	42.7
		CLAMP-ViT [62]	48.5	42.2
		SARDFQ (Ours)	**49.0**	**42.9**
	8/8	PSAQ-ViT V2 [48]	50.9	44.1
		PSAQ-ViT V2† [48]	51.0	44.2
		CLAMP-ViT [62]	51.4	**44.6**
		SARDFQ (Ours)	**51.7**	**44.6**

Table 7. Comparison of Mask R-CNN and Cascade R-CNN with Swin-S as the backbone. "†" indicates the results are re-produced by ours and the remaining results of PSAQ-ViT V2 [48] and CLAMP-ViT [62] are copied from their paper. "AP (box)" is the box AP for object detection and "AP (mask)" is the mask AP for instance segmentation.

Bit	Size (MB)	Speed (ms)	Running Memory (MB)
FP	346	534	1859
4-bit	43.3 (8×)	244 (2.2×)	1165 (1.6×)

Table 8. Comparison of FP and 4-bit ViT-B with 200 images.

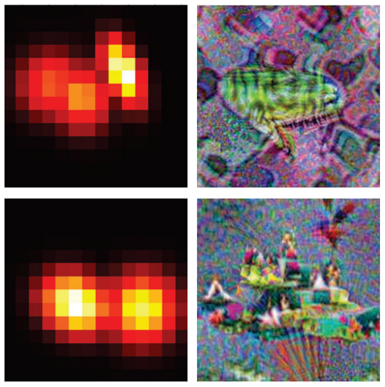

Figure 7. Attention priors (of the last layer) and generated images.

F.3. Runtime Examples

Performing data synthesis for ViT-B with SARDFQ on a 3090 GPU requires 13 minutes, while PSAQ-ViT and PSAQ-ViT V2 consume 8 and 25 minutes, respectively.