

Supplementary material for: Beyond Static and Dynamic Quantization - Hybrid Quantization of Vision Transformers

Piotr Kluska^{1,2}
klu@zurich.ibm.com

Florian Scheidegger¹
eid@zurich.ibm.com

A. Cristiano I. Malossi¹
acm@zurich.ibm.com

Enrique S. Quintana-Ortí²
quintana@disca.upv.es

¹ IBM Research Europe
Rüschlikon, Switzerland

² Universitat Politècnica de València
València, Spain

1 Hybrid Quantization algorithm

In this section, we provide details about the hybrid quantization algorithm. The Hybrid Quantization algorithm Algorithm 1 is based on the signal-to-quantization-noise ratio (SQNR) metric. We require a deep learning model ViT, calibration data, and a sample dataset to execute the algorithm. These two datasets are separate; one is used to calibrate the model, and the other is used to evaluate the sensitivity of the models to quantization.

First, we need to quantize the ViT model both dynamically and statically. The signal is then measured at the nodes of the linear layers N located between the quantized and reference models. Depending on the variant of the HQ algorithm, the signal is routed between the quantized model and the reference.

The HQ1 algorithm routes activations from the quantized model to the reference model. The HQ2 algorithm routes activations through individual networks. In HQ3, the activation of the reference model is quantized and propagated through the quantized model.

Finally, after selecting the layers, the model is calibrated using a calibration dataset and evaluated on the ImageNet1K validation dataset.

2 Additional experimental results

Here, we presented complementary experimental results of the hybrid quantization algorithms. In Table 1, we presented full ingestion of the best models attained with HQ and FQ-ViT methods. Additionally, in Table 2, we presented the mean and standard deviation out of the five runs that extend the results available in the main text and the Table 1. Figure 1 presented the latency measurements on the third environment - cloud CPU-only server

with Intel Xeon 5218 Gold. Finally, Figure 2 showed the static to dynamic linear layers quantization ratio.

Most of the models in the ViT family experienced low variability in top-1 accuracy when we applied hybrid quantization algorithms, as shown in Table 2. The exceptions were the ViT-B/16/224 and ViT-B/16/384 for HQ1 and HQ3, and the ViT-S/32/224 and ViT-S/32/384 for HQ3. Nevertheless, we observed an average speedup over dynamic quantization of 1.13

Algorithm 1 Post-training Hybrid Quantization algorithms. The specific operations in the HQ1, HQ2, and HQ3 algorithms are shown in the tables below. The remaining steps are common to all variants.

Require:

The dataset $D_s = \{s_1, s_2, \dots, s_i\}$ containing i samples
 The calibration dataset $D_c = \{s_1, s_2, \dots, s_j\}$ containing j calibration samples
 Neural network ViT = $\{\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_m\}$ consisting of m nodes
 Static quantization function \mathcal{Q}^S ; Dynamic quantization function \mathcal{Q}^D
 Dequantize function \mathcal{D} ; Calibration function calibrate

$\text{ViT}^S \leftarrow \mathcal{Q}^S(\text{calibrate}(\text{ViT}, D_c)); \text{ViT}^D \leftarrow \mathcal{Q}^D(\text{ViT})$

Initialize SQNR list L

for $s_i \in D_s$ **do**

$Y^S \leftarrow \mathcal{Q}^S(s_i); Y^D \leftarrow \mathcal{Q}^D(s_i); Y \leftarrow s_i$

for $\mathcal{N}_k \in \text{ViT}, \mathcal{N}_k^S \in \text{ViT}^S, \mathcal{N}_k^D \in \text{ViT}^D$ **do**

HQ1	HQ2	HQ3
$Z^S \leftarrow \mathcal{N}_k(\mathcal{D}(Y^S))$	$Y^S \leftarrow \mathcal{N}_k^S(Y^S)$	$Y^S \leftarrow \mathcal{N}_k^S(\mathcal{Q}^S(Y))$
$Y^S \leftarrow \mathcal{N}_k^S(Y^S)$	$Y^D \leftarrow \mathcal{N}_k^D(Y^D)$	$Y^D \leftarrow \mathcal{N}_k^D(\mathcal{Q}^D(Y))$
$Z^D \leftarrow \mathcal{N}_k(\mathcal{D}(Y^D))$	$Y \leftarrow \mathcal{N}_k(Y)$	
$Y^D \leftarrow \mathcal{N}_k^D(Y^D)$		

if \mathcal{N}_k is a linear layer **then**

HQ1	HQ2 & HQ3
$\omega^S \leftarrow \text{SQNR}^S(Z^S, Y^S)$	$\omega^S \leftarrow \text{SQNR}^S(Y, Y^S)$
$\omega^D \leftarrow \text{SQNR}^D(Z^D, Y^D)$	$\omega^D \leftarrow \text{SQNR}^D(Y, Y^D)$

Store $(\mathcal{N}_k, \omega^S, \omega^D)$ in L

end if

end for

end for

$L_G \leftarrow$ Group the list L by \mathcal{N}_k

for $\mathcal{N}_k, \omega^S, \omega^D \in L_G$ **do**

$\bar{\omega}^S \leftarrow \frac{1}{n} \sum_{i=1}^n \omega_i^S; \bar{\omega}^D \leftarrow \frac{1}{n} \sum_{i=1}^n \omega_i^D$

if $\bar{\omega}^S \geq \bar{\omega}^D$ **then**

Select \mathcal{N}_k for static quantization

else

Select \mathcal{N}_k for dynamic quantization

end if

end for

and 1.64 for ViT-B/16/224 and 1.23 and 1.65 for ViT-B/16/384 on an NVIDIA A100 GPU for the HQ1 and HQ3 algorithms, respectively. For the same models on a mobile A15 CPU, we achieved average speedups over dynamic quantization of 1.07 (HQ1), 1.21 (HQ3), 1.18 (HQ1), and 1.51 (HQ3); see Table 3 and Figure 1. Moreover, in Figure 4, we presented ViT-B/32/384 and ViT-L/32/384 latency versus accuracy trade-off plots. On the one hand, on average, we improved the latency of ViT-B/32/384 compared to static quantization. On the other hand, on average, except for the best model, the performance of the top-1 accuracy is degraded. HQ1 and HQ3 algorithms improved latency compared to dynamic quantization, with the best models for ViT-L/32/384 also achieved better accuracy than static quantization. In the ViT family models, we observed that the HQ2 algorithm preferred dynamic quantization over static linear layers. Meanwhile, HQ1 and HQ3 balanced the distribution of static and dynamic quantization; see Figure 2.

The DeiT models were the most robust to quantization. As a result, we achieved low variance in our evaluation while maintaining top-1 accuracy close to static quantization; see Table 2. Nevertheless, we improved the latency of the DeiT models compared to dynamic quantization by an average of 2.02 and 1.66 for HQ1 and HQ3, respectively, on a GPU-powered workstation; see Table 3.

Within the DeiT3 family models, we observed a small variance in most of the models, except for two outliers: DeiT3-B/16/224 and DeiT3-L/16/224; see Table 2 and Figure 3. However, within those models, we achieved up to 1.80 and 1.69 speedup compared to dynamic quantization on an NVIDIA A100 GPU. As for the mobile A15 CPU, we obtained up to 1.27 and 1.16 speedup compared to dynamic quantization. On average, we observed up to 1.33 and 1.72 for mobile A15 CPU and A100 GPU-powered workstations for the whole family of the models as presented in Table 3. We provided additional examples in Figure 4 of DeiT3-S/16/224 and DeiT3-B/16/384, where HQ1 and HQ3 achieved the best trade-offs of latency versus accuracy compared to dynamic and static quantization.

Finally, we observed similar robustness to quantization within the Swin Transformer models. Our algorithms for these models selected most static linear layers for quantization (see Figure 2), as they were found to be more robust to quantization errors than dynamically quantized linear layers.

Model	FP32	INT8	INT8-D	HQ1	HQ2	HQ3	FQ-ViT
ViT-T/16/224	75.47	9.02	71.66	9.27	12.41	13.18	45.80
ViT-T/16/384	78.42	7.65	74.98	9.90	11.05	10.55	45.18
ViT-S/16/224	81.39	77.01	79.34	77.12	73.43	77.05	78.58
ViT-S/16/384	83.80	79.40	82.46	76.81	77.08	79.77	81.61
ViT-S/32/224	75.99	45.59	74.38	60.93	59.43	60.58	59.13
ViT-S/32/384	80.48	59.77	79.06	71.84	71.62	71.25	68.43
ViT-B/16/224	85.10	73.15	84.15	71.76	70.27	74.89	83.70
ViT-B/16/384	86.00	51.05	85.31	48.82	48.75	51.78	84.11
ViT-B/32/224	80.71	71.35	79.13	75.10	74.99	74.57	70.99
ViT-B/32/384	83.35	81.32	82.70	81.44	80.49	81.16	81.37
ViT-L/16/224	85.85	84.30	85.48	83.94	84.18	84.40	84.97
ViT-L/32/384	81.51	80.62	81.36	80.91	80.35	80.75	81.36
DeiT-T/16/224	72.18	71.63	71.71	71.51	70.76	71.43	71.05
DeiT-T/16/224*	74.50	74.16	74.06	74.03	73.31	73.94	73.63
DeiT-S/16/224	79.85	78.84	79.14	78.70	78.32	78.74	78.49
DeiT-S/16/224*	81.22	80.75	80.79	80.86	80.66	80.68	80.42
DeiT-B/16/224	81.99	78.23	81.30	78.40	78.17	78.67	80.96
DeiT-B/16/224*	83.39	82.92	82.48	82.87	82.96	82.51	82.48
DeiT3-S/16/224	81.37	78.95	80.63	79.42	78.93	79.26	79.62
DeiT3-S/16/384	83.43	80.39	82.91	80.89	80.81	80.80	81.28
DeiT3-M/16/224	83.09	79.47	82.74	79.72	79.59	79.56	82.1
DeiT3-B/16/224	83.79	76.54	83.51	80.81	80.40	80.15	0.10
DeiT3-B/16/384	85.07	80.98	84.79	82.19	82.99	81.99	0.10
DeiT3-L/16/224	84.78	71.77	84.61	83.27	82.01	80.29	0.10
Swin-T/4/224	81.37	79.19	81.01	79.33	79.36	79.28	80.02
Swin-S/4/224	83.30	82.70	83.26	82.69	82.74	82.69	82.40
Swin-B/4/224	85.27	84.19	84.99	84.16	84.21	84.18	82.50
Swin-L/4/224	86.32	85.72	86.28	85.77	85.78	85.75	85.61

Table 1: Comparison between baseline (FP32), static quantization (INT8), dynamic quantization (INT8-D), hybrid quantization (HQ1, HQ2, and HQ3), and FQ-ViT. We report top-1 accuracy on the ImageNet1K dataset. In this experiment we present the best model out of 5 runs. We reproduce and extend the results of FQ-ViT. The superscript ((*)) in DeiT denotes the distilled version. In bold we mark the best result between HQ and FQ-ViT.

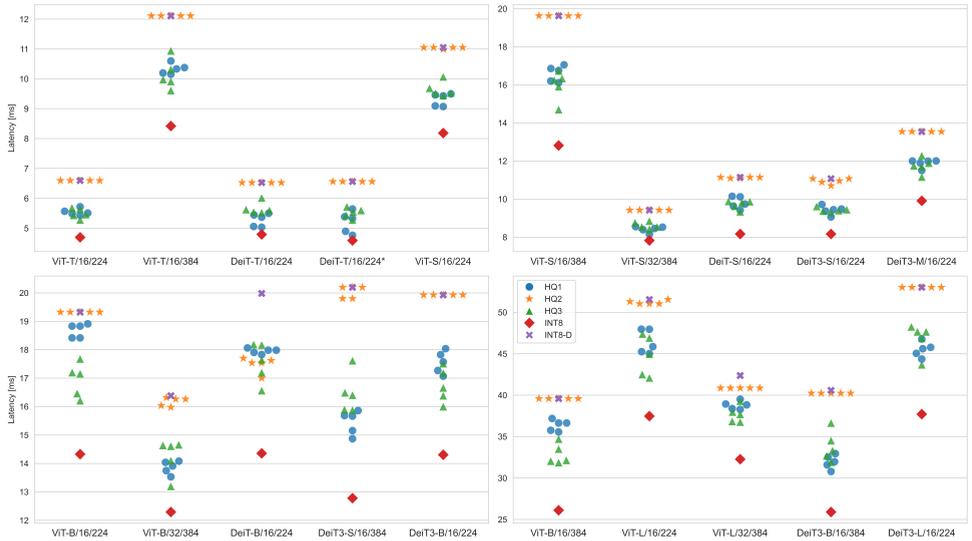


Figure 1: CPU-only workstation: Average Intel Xeon 5218 Gold CPU inference latency of models. Each point represents a single quantization configuration. We report the average latency of linear layers over 1000 samples.

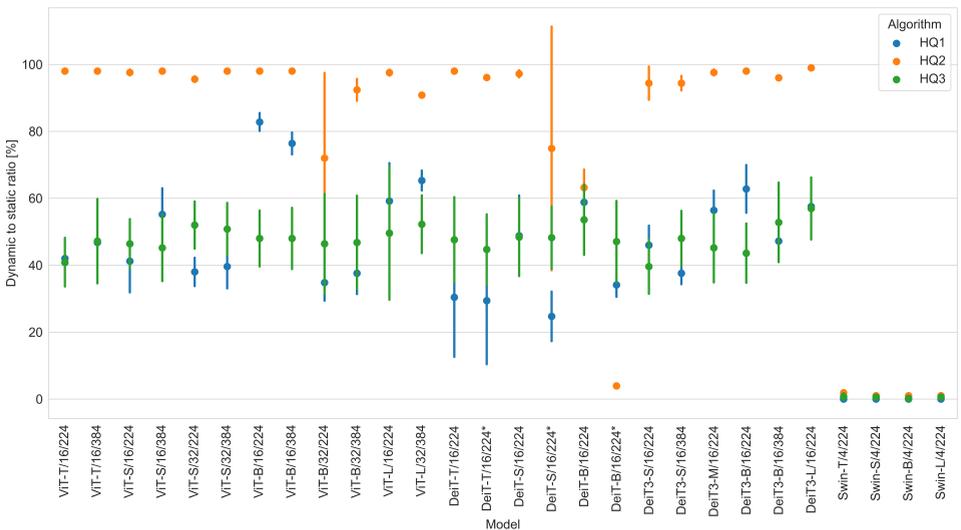


Figure 2: Percentage of dynamic quantized layers in models. The X-axis represents the model, while the Y-axis represents the ratio of dynamic to static layers. Higher values represent more dynamic layers in the model after quantization. Points represent the average of the ratio over five runs, with the standard deviation plotted.

Model	HQ1	HQ2	HQ3
ViT-T/16/224	8.32±0.62	12.25±0.13	10.84±1.51
ViT-T/16/384	7.75±1.52	10.01±0.82	8.22±1.75
ViT-S/16/224	76.80±0.21	73.12±0.21	74.70±1.20
ViT-S/16/384	76.60±0.20	76.99±0.08	78.63±1.31
ViT-S/32/224	60.26±0.47	59.24±0.15	54.27±7.19
ViT-S/32/384	71.35±0.35	71.35±0.19	64.25±5.73
ViT-B/16/224	68.64±2.38	69.81±0.34	71.59±1.87
ViT-B/16/384	43.83±4.17	46.17±1.90	48.00±2.23
ViT-B/32/224	73.81±1.35	74.00±1.07	72.65±1.33
ViT-B/32/384	81.06±0.30	80.16±0.23	80.68±0.39
ViT-L/16/224	83.80±0.09	84.01±0.14	84.19±0.24
ViT-L/32/384	80.68±0.14	80.18±0.11	80.45±0.32
DeiT-T/16/224	71.40±0.08	70.69±0.05	71.13±0.22
DeiT-T/16/224*	73.88±0.12	73.30±0.02	73.71±0.16
DeiT-S/16/224	78.41±0.15	78.18±0.14	78.53±0.15
DeiT-S/16/224*	80.78±0.07	80.21±0.23	80.48±0.14
DeiT-B/16/224	78.28±0.09	78.07±0.08	78.48±0.21
DeiT-B/16/224*	82.83±0.02	82.89±0.04	81.77±0.85
DeiT3-S/16/224	79.30 ± 0.12	78.77 ± 0.10	79.09 ± 0.12
DeiT3-S/16/384	80.78 ± 0.07	80.69 ± 0.07	80.68 ± 0.08
DeiT3-M/16/224	79.60 ± 0.10	79.55 ± 0.03	79.42 ± 0.12
DeiT3-B/16/224	61.78 ± 25.13	58.83 ± 24.69	70.58 ± 18.00
DeiT3-B/16/384	82.08 ± 0.09	81.62±1.39	81.69±0.22
DeiT3-L/16/224	72.26 ± 7.16	71.48 ± 7.52	73.07 ± 4.55
Swin-T/4/224	79.27±0.04	79.24±0.08	79.17±0.11
Swin-S/4/224	82.65±0.03	82.67±0.04	82.64±0.03
Swin-B/4/224	84.10±0.03	84.15±0.04	84.13±0.05
Swin-L/4/224	85.66±0.07	85.71±0.04	85.67±0.05

Table 2: Average top-1 accuracy with standard deviation on the ImageNet1K dataset computed over five runs. For DeiT models with ((*)), we denote the distilled version. Our method improves the top-1 accuracy over the reference INT8 static model in 12/12 ViT, 3/6 DeiT, 6/6 DeiT3, and 4/4 Swin models.

Model	A15 CPU			A100 GPU		
	HQ1	HQ2	HQ3	HQ1	HQ2	HQ3
ViT-T/16/224	1.17	1.00	1.20	1.90	1.00	1.88
ViT-T/16/384	1.39	1.00	1.43	1.71	1.00	1.66
ViT-S/16/224	1.23	1.00	1.21	1.84	1.00	1.70
ViT-S/16/384	1.35	1.00	1.48	1.49	1.00	1.71
ViT-S/32/224	1.41	1.00	1.28	2.14	1.03	1.63
ViT-S/32/384	1.21	1.00	1.16	1.92	1.00	1.62
ViT-B/16/224	1.07	1.00	1.21	1.13	1.00	1.64
ViT-B/16/384	1.18	1.00	1.51	1.23	1.00	1.65
ViT-B/32/224	1.36	1.13	1.32	1.97	1.26	1.72
ViT-B/32/384	1.22	1.02	1.18	1.99	1.05	1.74
ViT-L/16/224	1.16	1.01	1.22	1.41	1.02	1.64
ViT-L/32/384	1.08	1.03	1.12	1.40	1.14	1.64
ViT - Avg. all	1.23	1.01	1.28	1.68	1.04	1.68
DeiT-T/16/224	1.25	1.00	1.18	2.21	1.01	1.70
DeiT-T/16/224*	1.25	1.00	1.23	2.16	1.01	1.73
DeiT-S/16/224	1.17	1.00	1.18	1.66	1.00	1.66
DeiT-S/16/224*	1.43	1.61	1.25	2.51	1.22	1.66
DeiT-B/16/224	1.20	1.20	1.23	1.41	1.50	1.53
DeiT-B/16/224*	1.44	1.01	1.24	2.15	4.3	1.65
DeiT - Avg. all	1.29	1.15	1.21	2.02	1.67	1.66
DeiT3-S/16/224	1.22	1.01	1.22	1.81	1.00	1.87
DeiT3-S/16/384	1.74	1.01	1.50	1.91	1.01	1.67
DeiT3-M/16/224	1.17	1.00	1.20	1.55	1.00	1.76
DeiT3-B/16/224	1.17	1.00	1.27	1.47	1.00	1.80
DeiT3-B/16/384	1.50	1.01	1.42	1.88	1.00	1.57
DeiT3-L/16/224	1.16	1.00	1.15	1.69	1.00	1.49
DeiT3 - Avg. all	1.33	1.01	1.29	1.72	1.01	1.69

Table 3: The extended version of the average speedup latency improvement is over five runs of hybrid quantization compared to the dynamic quantization for ViT families for the iPhone 13 Pro smartphone with an A15 CPU and a workstation with an NVIDIA A100 GPU. In bold, we mark the best latency speedup for each environment.

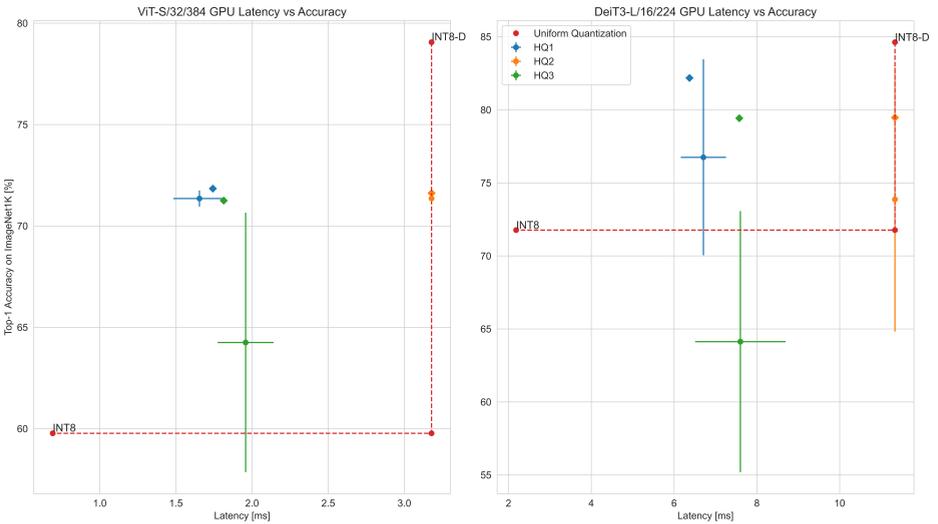


Figure 3: Latency vs accuracy trade-off of HQ algorithms compared to static quantization (INT8) and dynamic quantization (INT8-D) measured on an NVIDIA A100 GPU. We plot the mean and standard deviation for all the runs. The point marked with a diamond corresponds to the model that offers highest accuracy. The results above and left to the dotted red line indicate improvement in accuracy compared to static quantization and latency compared to dynamic quantization, respectively.

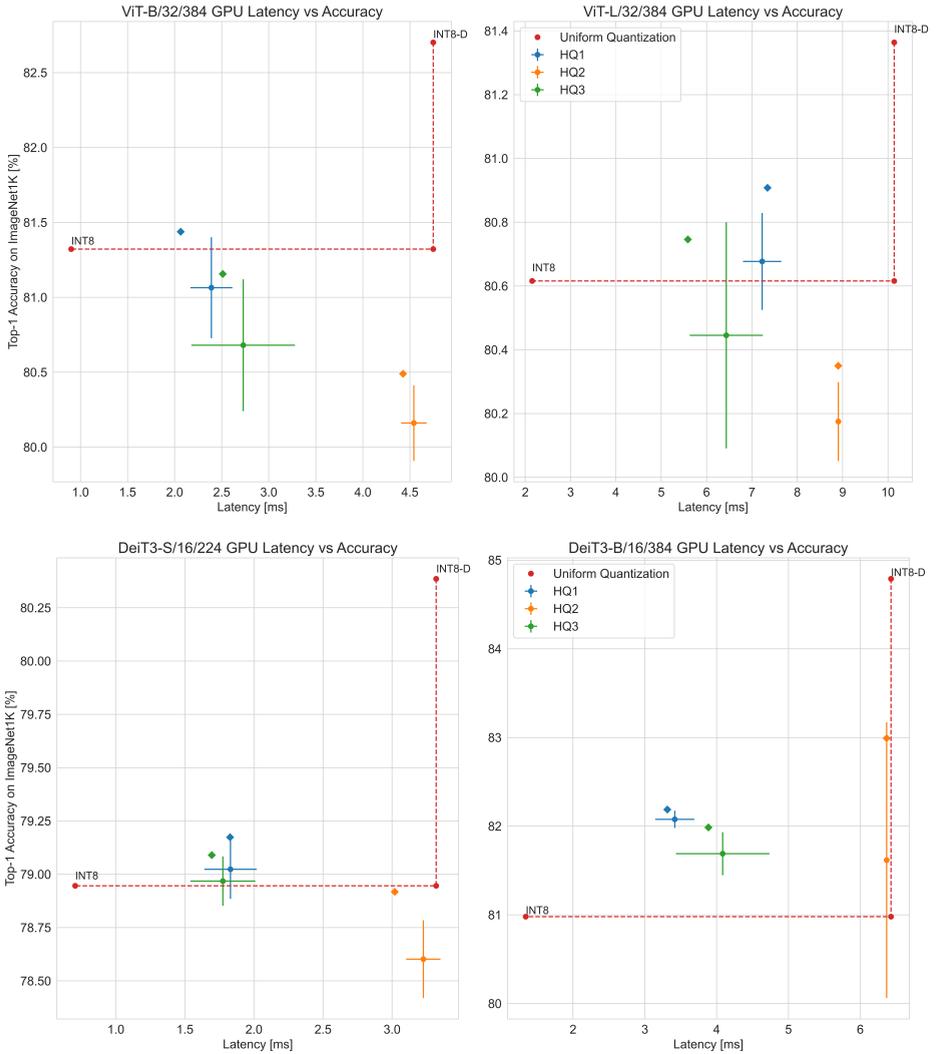


Figure 4: Latency vs accuracy trade-off of HQ algorithms compared to static quantization (INT8) and dynamic quantization (INT8-D) measured on an NVIDIA A100 GPU. We plot the mean and standard deviation for all the runs. The point marked with a diamond corresponds to the model that offers highest accuracy. The results above and left to the dotted red line indicate improvement in accuracy compared to static quantization and latency compared to dynamic quantization, respectively.