Quant Experts: Token-aware Adaptive Error Reconstruction with Mixture of Experts for Large Vision-Language Models Quantization

Supplementary Material

In the supplementary material, we provide additional Related Work, Method Details, and Experimental Results. In Sec. 7, we present more complete implementation details for the Shared Expert and the Refinement of Routed Experts. In Sec. 8, we report the main results of QE on Qwen2VL-7B and InternVL2-2B, along with evaluations on language tasks. We further present results for joint quantization of the Visual Encoder and the VLM, along with an extended ablation study on the Number of Important Channels.

6. Additional Details for Related Work

In large language model (LLM) compression, two mainstream approaches are commonly used: quantization-aware training (QAT) and post-training quantization (PTQ). QAT explicitly models quantization errors during training and can achieve higher accuracy for low-bit models, but it incurs substantial computational and data overhead (*e.g.*, LSQ [10], LLM-QAT [26], DL-QAT [17]). In contrast, PTQ directly maps pretrained weights and activations into low-bit representations after training, requiring only a small amount of calibration data. Due to its efficiency and practicality, PTQ has become the dominant solution for resource-constrained scenarios (*e.g.*, QBB [3], aespa [20]). However, PTQ inevitably introduces quantization errors, and existing methods remain constrained by limited outlier identification and error compensation, posing a key challenge for advancing low-bit LLM deployment [13].

To address this core challenge, various solutions have been proposed from different perspectives. OBQ [11] and GPTQ [12] perform progressive quantization with Hessian-guided iterative compensation, allowing unquantized parameters to absorb the quantization errors yielded in previous channels or blocks, thereby alleviating reconstruction error within Transformer blocks. [39] further combines Hessian-based optimization with the Expectation-Maximization (EM) algorithm to enable joint weight-activation quantization at extremely low bitwidths. Distribution reshaping approaches mitigate the effects of outliers by applying channel-wise scaling and equalization to balance the dynamic ranges of activations and weights. Among them, SmoothQuant [44] transfers part of the quantization difficulty from activations to weights through channel-wise scaling, effectively balancing their dynamic ranges. Furthermore, AWQ [24] employs a search-based channel scaling strategy, while selectively retaining the most sensitive parameters in full precision to preserve model accuracy. OmniQuant [35] incorporates learnable clipping and equivalent scaling transformations, jointly optimized under a block-level error minimization framework to achieve stronger error suppression. From another perspective, some approaches utilize rotation-based transformations to mitigate outliers in weight and activation quantization, effectively reducing quantization errors. QuIP [4] employs an uncorrelated transformation combined with adaptive rounding to minimize proxy errors, while QuIP# [41] integrates random Hadamard transforms and block-wise vector quantization to improve reconstruction accuracy. QuaRot [1] further proposes an end-to-end 4-bit quantization scheme based on Hadamard rotation, which enables simultaneous quantization of weights, activations, and KV cache. In model quantization, performance degradation and quantization errors primarily arise from outlier and sensitivity-prone important channels. Precisely identifying and preserving these channels at higher precision is essential for mitigating quantization errors. For instance, Atom [51] enhances robustness under low-bit settings through hybrid precision and dynamic activation quantization, whereas SpQR [9] leverages Hessian-based sensitivity analysis to identify important parameters, retaining high precision for outlier weights while quantizing the remaining ones into low-bit representations, thereby effectively mitigating outlier-induced errors. Another research direction introduces low-rank structures into quantization error compensation by attaching lightweight high-precision low-rank modules to recover accuracy with minimal computational and memory overhead. Representative approaches include LoRC [45], which models quantization residuals using low-rank matrices to restore performance at low cost; LQER [48], which leverages activation statistics and diagonal rescaling for weighted low-rank reconstruction; and ASER [50], which adopts whitened SVD for more stable error modeling and integrates outlier-channel analysis to smooth activation distributions.

7. Additional Details for Method

7.1. Shared Expert

The construction process of the shared expert in QE is illustrated in Algorithm 3, which follows the general procedure described in [50]. This method employs a low-rank structure to approximate the quantization error introduced by weight quantization, with a particular focus on frequently activated, token-independent important channels, thereby

effectively capturing globally stable quantization error patterns.

Algorithm 3: Building the Shared Expert

Input : Per-layer data $\{\mathbf{X}^l,\ \mathbf{W}_f^l,\ \mathcal{C}_s^l\}_{l=1}^L$, quantizer $Q(\cdot)$; rank r.
Output: Quantized layer weight $\{\mathbf{W}_q^l\}_{l=1}^L$, SE $\{(\mathbf{L}_{SA}^l, \mathbf{L}_{SB}^l)\}_{l=1}^L$, and residual errors $\{\mathbf{E}_S^l\}_{l=1}^L$.

Compute $\mathbf{x}^l \leftarrow \mathrm{Mean}_{\text{row}}(|\mathbf{X}^l|)$
for $l \leftarrow 1$ **to** L **do**
  Initialize $\omega = [1, 1, \ldots, 1]^n$, $\Omega = \mathrm{diag}(\omega)$
  Compute $\omega_{\mathcal{C}_s^l} = \mathbf{x}_{\mathcal{C}_s^l}^l / \min(\mathbf{x}_{\mathcal{C}_s^l}^l)$
  $\mathbf{E}_q^l = \mathbf{W}_f^l - Q(\mathbf{W}^l \mathrm{diag}(\mathbf{1} - \mathbf{1}_{\mathcal{C}_s^l}))$
  Compute whitening matrix S by Cholesky decomposition of $(\Omega^{-1}X)(\Omega^{-1}X)^\top$ such that $(S^{-1}\Omega^{-1}X)(S^{-1}\Omega^{-1}X)^\top = I$
  Perform SVD: $U\Sigma V^\top = E_q^l S$
  Compute: $\mathbf{L}_{SA}^l = U_r\Sigma_r$, $\mathbf{L}_{SB}^l = V_r^\top S^{-1}$, $\mathbf{E}_S^l = \mathbf{E}_q^l - \mathbf{L}_{SA}^l\mathbf{L}_{SB}^l$
return $\{\mathbf{W}_q^l\}_{l=1}^L$, $\{(\mathbf{L}_{SA}^l, \mathbf{L}_{SB}^l)\}_{l=1}^L$, $\{\mathbf{E}_S^l\}_{l=1}^L$

7.2. Refinement of Routed Experts

In this subsection, we describe the loss functions used in the Refinement stage. These losses follow standard formulations commonly adopted in prior research. We provide detailed explanations here due to space limitations in the main paper. The refinement objective consists of two complementary losses: a regression loss $\mathcal{L}_{\text{reg}}$ and a classification loss $\mathcal{L}_{\text{cls}}$. $\mathcal{L}_{\text{reg}}$ aims to minimize the reconstruction error between the quantized output $\hat{y}$ and full-precision output y, encouraging each expert to specialize in its own direction of compensation. $\mathcal{L}_{\text{cls}}$ improves the router's ability to predict the optimal expert for a given input.

Specifically, let y_i denote the output reconstructed by the i-th routed expert and y the full-precision output. We define the reconstruction distance as $d_i = \|\hat{y}_i - y\|_1$. During refinement, only the routed expert achieving the smallest reconstruction error is optimized, formulated as:

$$\mathcal{L}_{\text{reg}} = \min_{i \in [1, N_r]} d_i. \tag{11}$$

To enable the router to predict the relative performance of different routed experts, we denote its output as $l = R|x|$ and construct a classification objective based on the inter-expert discrepancy. We adopt the Kullback-Leibler divergence to align the predicted distribution with the normalized reconstruction loss distribution:

$$\mathcal{L}_{\text{cls}} = \tau^2 D_{\text{KL}}\left(\mathbf{P} \parallel \mathbf{Q}\right), \tag{12}$$

$$\mathbf{P} = \mathrm{softmax}\left(\frac{-(d - \mu(d))/\sigma(d)}{\tau}\right), \tag{13}$$

$$\mathbf{Q} = \mathrm{softmax}\left(\frac{-(l - \mu(l))}{\tau}\right), \tag{14}$$

where τ is a temperature coefficient, and $\mu(\cdot)$ and $\sigma(\cdot)$ denote the mean and standard deviation. Finally, we use two coefficients, α and β, to balance the two losses:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{reg}} + \beta\mathcal{L}_{\text{cls}}. \tag{15}$$

8. Additional Experiments

8.1. Additional Model

The experimental results of QE on additional models are presented in Tab. 9 and Tab. 10. Consistent with previous findings, our method significantly outperforms the baselines under W4A6, W4A8, and W3A16 configurations.

8.2. Performance on Language Tasks

The core idea of QE is to employ multi-expert low-rank adapters that dynamically adapt to compensation differences across modalities and even among individual tokens, thereby improving model performance on both vision-language and language-only tasks. To validate this, we evaluate the quantized Qwen2VL-2B and Qwen2VL-7B models on the MMLU benchmark under different quantization methods. As shown in Tab. 11, QE consistently achieves significant performance gains over LQER across various quantization configurations and model scales. These results demonstrate that explicitly modeling sensitivity differences across modalities and tokens not only effectively mitigates performance degradation in vision-language tasks but also helps maintain stable performance on language-only tasks.

8.3. Quantize Both Visual Encoder and VLM

To achieve higher acceleration ratios, we further quantize both the visual encoder and the merger module that connects the encoder to the VLM. As shown in Tab. 12, L denotes the VLM, V the visual encoder, and M the merger module, where ✓ indicates that the corresponding module is quantized. The results show that as more modules are quantized, the model exhibits negligible performance degradation, demonstrating that the proposed joint quantization strategy effectively improves overall efficiency while maintaining accuracy.

8.4. Effect of the Number of Important Channels

We further investigate the effect of the number of important channels k on model accuracy in Tab. 13. The results

Method	#W	#A	MMMU	OCRBench	ScienceQA	TextVQA	VizWiz	AI2D	ChartQA	DocVQA	InfoVQA	MMStar	MuriBench	Avg. (↑)
Qwen2VL-7B	16	16	50.78	79.50	84.83	81.48	68.56	80.60	81.68	91.68	69.77	57.74	42.92	71.78
RTN	4	6	39.00	59.50	75.41	66.93	56.64	69.62	72.00	78.73	52.91	49.97	38.27	59.91
SQ (ICML'23)	4	6	41.00	63.90	77.09	67.94	57.48	70.92	68.84	78.78	52.90	50.48	35.65	60.45
LQER (ICML'24)	4	6	42.56	65.60	77.24	**71.87**	**64.44**	71.44	74.04	81.57	56.86	49.75	41.85	63.38
MBQ (CVPR'25)	4	6	40.56	62.70	79.67	70.91	51.48	71.31	72.20	81.99	54.71	47.37	34.54	60.68
QE	4	6	**45.44**	**73.00**	**79.87**	71.63	64.18	**75.58**	**74.16**	**83.60**	**60.45**	**52.63**	**42.19**	**65.70**
RTN	4	8	45.44	60.30	79.47	71.18	59.11	76.78	74.52	77.04	56.89	53.23	40.04	63.09
SQ (ICML'23)	4	8	43.78	58.60	79.52	69.52	53.01	76.20	72.88	74.34	53.95	52.93	36.69	61.04
LQER (ICML'24)	4	8	**48.00**	69.50	**81.76**	75.31	66.02	77.85	77.04	82.27	61.52	**55.24**	**43.85**	67.12
MBQ (CVPR'25)	4	8	46.33	72.00	81.46	75.35	60.78	76.98	76.32	85.26	61.62	53.52	37.65	66.12
QE	4	8	46.33	**78.20**	81.51	**78.98**	**66.59**	**79.05**	**78.92**	**89.21**	**66.16**	54.04	42.46	**69.22**
RTN	3	16	32.44	65.80	67.63	70.43	55.81	76.75	73.72	74.46	58.14	50.32	43.69	60.84
AWQ (MLSys'24)	3	16	**48.00**	76.30	82.15	79.00	65.69	76.49	78.92	89.10	64.98	**54.01**	40.69	68.67
LQER (ICML'24)	3	16	46.44	64.50	80.81	72.50	66.22	76.91	75.36	77.42	59.77	52.01	43.27	65.02
MBQ (CVPR'25)	3	16	46.22	74.40	**82.35**	79.43	65.02	77.30	78.24	88.59	64.70	52.45	42.96	68.33
QE	3	16	46.67	**77.20**	81.56	**79.87**	**67.20**	**78.01**	**79.28**	**89.60**	**65.26**	53.35	**44.58**	**69.33**

Table 9. Main results on the model of Qwen2VL-7B.

Method	#W	#A	MMMU	OCRBench	ScienceQA	TextVQA	VizWiz	AI2D	ChartQA	DocVQA	InfoVQA	MMStar	MuriBench	Avg. (↑)
InternVL2-2B	16	16	34.33	75.30	94.30	72.58	45.94	72.83	74.84	84.84	53.23	48.20	28.46	62.26
RTN	4	6	30.44	67.10	86.47	66.17	41.66	63.41	66.48	77.41	43.91	40.11	26.85	55.46
SQ (ICML'23)	4	6	31.89	69.30	88.25	67.24	40.17	64.41	65.48	79.94	46.94	41.62	25.69	56.45
LQER (ICML'24)	4	6	30.78	70.90	88.35	67.81	39.33	65.64	68.92	79.78	45.73	41.75	28.27	57.02
MBQ (CVPR'25)	4	6	31.33	70.90	90.53	68.54	41.39	67.52	70.20	80.99	47.95	**45.26**	25.46	58.19
QE	4	6	**32.11**	**72.80**	**92.12**	**70.41**	**43.81**	**68.69**	**70.52**	**82.00**	**48.69**	45.18	**28.69**	**59.55**
RTN	4	8	32.00	72.10	91.08	69.19	42.72	68.07	69.04	81.06	48.97	45.12	28.27	58.87
SQ (ICML'23)	4	8	33.78	71.20	91.27	69.20	40.19	68.13	68.04	81.31	48.75	44.23	26.69	58.44
LQER (ICML'24)	4	8	**34.56**	72.50	92.07	70.53	39.16	69.33	70.96	82.03	49.89	44.84	28.65	59.50
MBQ (CVPR'25)	4	8	32.78	72.50	92.22	70.19	**44.31**	70.53	71.44	82.24	49.89	**47.73**	27.31	60.10
QE	4	8	32.33	**74.00**	**92.86**	**71.44**	43.29	**71.47**	**72.88**	**83.30**	**51.11**	45.77	**29.08**	**60.68**
RTN	3	16	29.78	69.70	88.65	67.51	38.21	66.09	68.88	80.56	46.29	41.45	28.38	56.86
AWQ (MLSys'24)	3	16	29.78	69.50	89.89	68.12	45.30	67.78	68.44	80.84	46.44	44.68	25.50	57.84
LQER (ICML'24)	3	16	**31.00**	70.30	89.34	67.80	35.93	67.03	69.52	80.41	46.57	41.74	28.04	57.06
MBQ (CVPR'25)	3	16	30.33	69.20	89.39	67.83	45.74	67.68	68.40	80.57	46.21	44.20	26.27	57.80
QE	3	16	30.78	**72.10**	**92.76**	**69.60**	**47.68**	**70.05**	**71.44**	**82.16**	**47.90**	**45.08**	**29.77**	**59.94**

Table 10. Main results on the model of InternVL2-2B.

Model	Setting	Method	MMLU (↑)
Qwen2VL-2B	FP16	-	52.79
	W4A6	LQER	44.37
		QE	**47.21**
	W4A8	LQER	46.60
		QE	**50.35**
Qwen2VL-7B	FP16	-	67.88
	W4A6	LQER	55.59
		QE	**61.87**
	W4A8	LQER	64.21
		QE	**64.83**

Table 11. The results of quantized Qwen2VL on the MMLU benchmark.

indicate a steady improvement as k increases. However, at $k = 64$, the performance saturates and slightly declines, as selecting an excessively large set of channels dilutes the focus on truly critical ones.

L	V	M	OCRBench	ScienceQA	TextVQA	VizWiz	Avg. (↑)
-	-	-	74.90	76.95	77.72	65.73	73.83
✓	-	-	68.20	71.84	73.18	59.62	68.21
✓	✓	-	65.90	70.75	72.09	59.83	67.14
✓	✓	✓	66.40	70.10	71.77	59.28	66.89

Table 12. Quantization results of different modules in Qwen2VL-2B. The symbol "-" indicates full precision (FP16), while ✓ denotes W4A6 quantization. L, V, and M correspond to the VLM, visual encoder, and merger module, respectively.

k	MMMU	OCRBench	VizWiz	Avg. (↑)
4	34.89	67.30	58.93	53.71
8	35.33	68.90	58.95	54.39
16	35.11	68.50	59.92	54.51
32	36.11	69.30	60.24	55.22
64	34.44	70.10	59.57	54.70

Table 13. Impact of the number of important channels on the performance of Qwen2VL-2B under the W4A6 quantization setting.