Supplementary Material
QuEPT: Quantized Elastic Precision Transformers with One-Shot Calibration for Multi-Bit Switching

Anonymous submission

LoRA Meets Post-Training Quantization

Although prior work has attempted to use LoRA as learnable parameters in quantization, the majority of these efforts have focused on combining LoRA with downstream task fine-tuning or Quantization-Aware Training (QAT) (Dettmers et al. 2023; Li et al. 2024; Xu et al. 2024b; Yelysei et al. 2024). In our work, we conduct a preliminary exploration of how to train LoRA more effectively in Post-Training Quantization (PTQ) and have performed some validation experiments. We found that during the process of learning LoRA in PTQ, the size of the optimization space and the resulting performance differ significantly depending on the stage at which LoRA is used to compensate for quantization. We broadly classify these approaches into three types:

- **Type 1: Weight-Space LoRA.** QuEPT utilizes Weight-Space LoRA to train its low-rank compensation parameters. Since LoRA operates directly on the original weight space, this approach provides a larger optimization space for training. Furthermore, it introduces no extra inference overhead because the LoRA weights can be merged back into the original weight space after PTQ. This method can be written as:

$$\hat{\boldsymbol{W}} = \boldsymbol{s_w} \cdot clip(\lfloor \frac{\boldsymbol{W}+\boldsymbol{BA}}{\boldsymbol{s_w}} \rceil, -2^{b-1}, 2^{b-1}-1). \quad (1)$$

- **Type 2: Scale-Space LoRA.** LR-QAT (Yelysei et al. 2024) experimented with this method of combining LoRA with Quantization-Aware Training (QAT) on large language models. Since LoRA is optimized in the space after division by the scale factor, we argue that this approach has a more constrained optimization space compared to Type 1. It also introduces no extra inference overhead. The formula for this method is as follows:

$$\hat{\boldsymbol{W}} = \boldsymbol{s_w} \cdot clip(\lfloor \frac{\boldsymbol{W}}{\boldsymbol{s_w}} + \boldsymbol{BA} \rceil, -2^{b-1}, 2^{b-1}-1). \quad (2)$$

- **Type 3: Rounding-Policy LoRA.** CBQ (Ding et al. 2025) adopts a method where the parameter-heavy rounding matrix of AdaRound (Nagel et al. 2020) is replaced with LoRA for training the rounding policy on large models. This approach reduces the training overhead of the rounding method; however, its performance is upper-bounded by that of the original AdaRound. Due to the additional regularization terms in the optimization objective, this approach has a slower optimization speed and greater memory overhead compared to Type 1 and Type 2. This method can be expressed by the following formula:

$$\hat{\boldsymbol{W}} = \boldsymbol{s_w} \cdot clip(\left\lfloor \frac{\boldsymbol{W}}{\boldsymbol{s_w}} \right\rfloor + \Delta\boldsymbol{W}, -2^{b-1}, 2^{b-1}-1), \quad (3)$$

$$\Delta\boldsymbol{W} = clip(Sigmoid(\boldsymbol{BA})(\zeta-\gamma)+\gamma, 0, 1), \quad (4)$$

$$\mathcal{L}_{\text{reg}} = \sum_{i,j} \left(1 - |2\cdot\Delta\boldsymbol{W}_{ij} - 1|^{\beta}\right). \quad (5)$$

The overall loss function is defined as follows:

$$\min_{\boldsymbol{A},\boldsymbol{B}} \left\| \boldsymbol{W}\boldsymbol{X} - \hat{\boldsymbol{W}}\hat{\boldsymbol{X}} \right\| + \alpha\mathcal{L}_{\text{reg}}, \quad (6)$$

where ζ and γ are stretch parameters and are fixed to 1.1 and -0.1, and $clip()$ clamps the inputs into a given range. We compared the performance of these three methods on ViT-S, DeiT-S, Swin-S, and LLaMA2-7B. We also divide [4,5,6,7,8] into high [7,8], mid [5,6], and low [4] groups, each group share $r = 16$ LoRA. For the LLaMA2-7B model, we compare the average accuracy across four datasets: PIQA, ARC-C, ARC-E, and Hellaswag.

As shown in the Table 1, entries marked with an asterisk (*) denote our reproductions implemented within our framework using the LoRA approaches described above. The results indicate that the Type 1 LoRA method achieves the best performance. Our reproduced results for Type 3 were suboptimal. We argue that Type 1 (QuEPT) outperforms Type 2 (LR-QAT*) because it has a larger optimization space. Meanwhile, the poor performance of our reproduced Type 3 (CBQ*) is likely attributable to LoRA's initialization. In our Type 3 implementation, matrix A is randomly initialized while B is set to zero, whereas the original AdaRound allows for a better initialization, thus providing a more favorable starting point for optimization. The initialization of LoRA is speculated to be a potential reason for our inability to replicate the results.

Additional Experiments

MB-ToMe Hyperparameter Sensitivity Analysis.

We tested the impact of selecting the top p% of tokens with MB-ToMe on ViT-S, as illustrated in Figure 1. As p increases, the performance of W4A4 is significantly affected.

Model	Method	**W4A4**	**W5A5**	**W6A6**	**W7A7**	**W8A8**
ViT-S	LR-QAT* (Type 2)	72.68	78.76	80.27	80.80	81.04
	CBQ* (Type 3)	0.10	23.98	47.93	78.04	80.57
	QuEPT (Type 1)	**74.81**	**79.23**	**80.41**	**80.93**	**81.11**
	AdaRound*	40.81	50.75	69.80	78.61	80.39
DeiT-S	LR-QAT* (Type 2)	74.42	77.95	79.15	**79.62**	79.68
	CBQ* (Type 3)	0.10	31.47	54.96	74.43	78.69
	QuEPT (Type 1)	**75.34**	**78.19**	**79.17**	79.57	**79.68**
	AdaRound*	30.10	56.00	69.99	76.91	78.93
Swin-S	LR-QAT* (Type 2)	81.80	82.59	82.94	**83.12**	83.16
	CBQ* (Type 3)	57.98	74.15	80.92	82.55	83.09
	QuEPT (Type 1)	**81.89**	**82.66**	**82.98**	83.09	**83.19**
	AdaRound*	80.23	81.42	82.09	82.71	83.09
LLaMA2-7B	LR-QAT* (Type 2)	60.73	64.17	64.51	64.77	65.15
	CBQ (Type 3)	50.84	-	60.97	-	-
	QuEPT (Type 1)	**61.66**	**64.27**	**64.88**	**65.26**	**65.96**

Table 1: Comparison of various LoRA-based PTQ methods. LR-QAT* and CBQ* are our reproduced results based on the LoRA approaches in their respective papers. AdaRound* is our reproduced result under the same experimental setup. CBQ for LLaMA2-7B shows the average accuracy across four datasets (PIQA, ARC-C, ARC-E, Hellaswag) from the original paper.

This is because an excess of 8-bit features leads to large reconstruction errors for 4-bit quantization. In contrast, the performance for medium-to-high bit-widths shows a slight improvement, because the feature distribution obtained from MB-ToMe becomes more similar to that of high-bit representations.

Furthermore, we present the effect of the mixing ratio λ on the results when there is a large discrepancy between high-bit-width features and low-bit-width group features, as shown in Table 2. When the mixing ratio is relatively uniform, such as 0.333:0.333:0.333 or 0.3:0.3:0.4, the overall performance across all bit-widths is more balanced. Conversely, if the ratio for a specific bit-width group is higher, the final test result will be biased towards that particular bit-width.

Different Bit-Groups Combinations.

As shown in Table 3, we tested the effect of partitioning the bit-widths [4,5,6,7,8] into three groups in different ways. The partitioning scheme [4][5,6][7,8] performs better across most bit-widths, particularly for the W4A4 results. The best performance is achieved by isolating [4] into its own group; in other words, partitioning the lowest bit-width separately is most effective. This is because the lowest bit-width typically requires more LoRA to compensate for quantization error. This principle can also be applied to weight-only quantization. It is worth noting that our method is still insufficient for extremely low-bit scenarios. For instance, at W3A3, the performance of other bit-widths is also negatively affected and degrades.

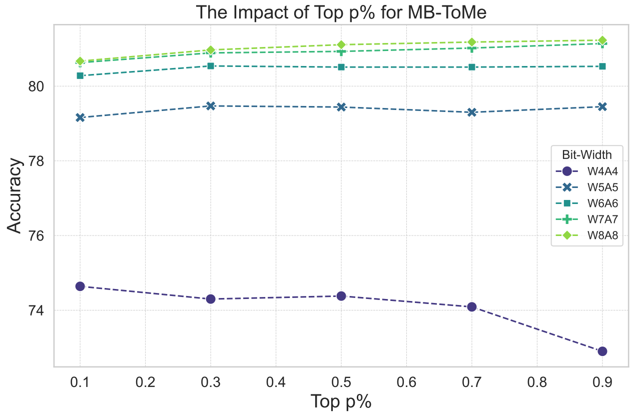

Figure 1: The hyperparameter sensitivity of MB-ToMe's top p% on ViT-S.

$\lambda_1 : \lambda_2 : \lambda_3$	**W4A4**	**W5A5**	**W6A6**	**W7A7**	**W8A8**
0.333:0.333:0.333	74.25	79.44	80.59	80.94	81.14
0.3:0.3:0.4	74.10	79.44	80.52	80.93	81.11
0.2:0.2:0.6	74.36	79.15	80.27	80.79	80.89
0.1:0.1:0.8	74.29	79.05	80.02	80.50	80.60
0.8:0.1:0.1	73.81	79.43	80.47	81.12	81.26

Table 2: The impact of λ in MB-ToMe on ViT-S.

LoRA Rank Ablation Study

We tested the impact of increasing the LoRA rank for each bit-width group on ViT-S, with the results shown in Table 4. As the LoRA rank rises, there is a significant improvement in the performance of low-bit configurations, especially W4A4. The performance gradually saturates when the rank reaches around 64. In contrast, other bit-widths are not significantly affected by the rank, because medium-to-

Model	Group	W3A3	W4A4	W5A5	W6A6	W7A7	W8A8
ViT-S	[4,5],[6],[7,8]	-	73.42	78.82	**80.46**	80.81	80.93
	[4,5,6],[7],[8]	-	72.04	78.73	79.97	80.68	80.98
	[4],[5,6,7],[8]	-	73.45	78.29	79.14	79.54	79.62
	[4],[5,6],[7,8]	-	**74.39**	**79.20**	80.41	**80.93**	**81.10**
	[3],[4,5],[6,7,8]	24.77	71.79	77.33	79.40	79.88	80.05
DeiT-S	[4,5],[6],[7,8]	-	73.45	78.01	79.14	79.54	79.62
	[4,5,6],[7],[8]	-	73.09	**78.10**	79.07	**79.69**	**79.77**
	[4],[5,6,7],[8]	-	74.49	77.98	**79.20**	79.51	79.70
	[4],[5,6],[7,8]	-	**74.51**	78.06	79.08	79.64	79.74
	[3],[4,5],[6,7,8]	44.82	73.02	77.53	78.90	79.33	79.43

Table 3: Result of different bit-Groups combinations. We validate Top-1 accuracy on ImageNet.

high bit-widths do not require an excessive rank to compensate for quantization error.

Rank	W4A4	W5A5	W6A6	W7A7	W8A8
1	73.28	79.36	80.42	81.06	81.12
2	73.84	79.23	80.53	80.89	81.12
4	74.10	79.44	80.51	80.93	81.11
8	74.67	79.47	80.36	80.98	81.11
16	75.12	79.61	80.49	80.97	81.16
32	75.53	79.42	80.37	80.91	81.12
64	**75.70**	79.42	80.41	80.93	81.17
128	75.69	79.46	80.24	80.93	81.05

Table 4: LoRA rank ablation study on ViT-S model.

Model	Bits	Runtime(ms)
LLaMA2-7B	FP16	2.57
	INT4	0.75
	INT4+QuaRot	0.80
LLaMA2-13B	FP16	3.98
	INT4	1.06
	INT4+QuaRot	1.14

Table 5: Performance of 16-bit and 4-bit linear layer for 2048 sequence lengths with and without QuaRot's online Hadamard transformation on a NVIDIA RTX 3090 GPU, averaged over 1000 runs. The matrix sizes correspond to the linear layer sizes in LLAMA-2 FFN blocks (i.e.W_{down}). Here the batch size is 1.

Weight-Only Quantization of LLaMA Series Models.

For 2-bit weight-only quantization on LLaMA, we, along with baselines (AWQ (Lin et al. 2024), GPTQ (Frantar et al. 2023), OmniQuant (Shao et al. 2024), Any-Precision-LLM (Park et al. 2024)), use per-group quantization with a group size of 128. For all other bit-widths, per-channel quantization is applied to all methods. The results in Table 11 demonstrate the robustness of our method's performance across various bit-widths. To demonstrate the robustness and effectiveness of our method, we perform extensive weight-only quantization experiments on a range of LLaMA family models, including LLaMA1-7B, LLaMA2-7B/13B, and LLaMA3-8B. As presented in Table 11, we compare QuEPT against leading single-bit quantization methods such as GPTQ, AWQ, and OmniQuant, across 2, 3, 4, and 6-bit weight precision. The results highlight QuEPT's significant superiority, especially in the ultra-low bit regimes (W2 and W3), where it drastically outperforms all single-bit baselines. For instance, in the challenging W2 setting on LLaMA2-13B, QuEPT achieves a perplexity of 7.89 on

Model	Batch Size	Speedup
LLaMA2-7B	1	1.97×
	4	2.06×
	16	2.11×
	32	2.14×
	64	2.16×

Table 6: Time-to-first token (prefill) speedup of each transformation block of LLAMA-2-7B models in QuEPT+QuaRot (over the FP16 model) on NVIDIA RTX3090 GPU.We use 2048 sequence lengths with different batch sizes.

Model	Bits	Memory(GB)	Saving
LLaMA2-7B	FP16	13.0	-
	INT8	6.5	2.00×
	INT7	5.8	2.24×
	INT6	5.0	2.60×
	INT5	4.7	3.25×
	INT4	3.9	3.33×
	INT3	3.1	4.19×
	INT2	2.3	5.65×

Table 7: Weight-only quantization file size.

LLaMA-7B		4 bit			3.x bit			3 bit			2.x bit	
	Bit	Wiki	C4	Bit	Wiki	C4	Bit	Wiki	C4	Bit	Wiki	C4
FP16	16	5.68	7.08	16	7.08	5.68	16	5.68	7.08	16	5.68	7.08
SqueezeLLM	4	5.79	7.21	3.24	6.13	7.56	3	6.32	7.75	2.23	11.32	15.69
OmniQuant	4	5.86	7.34	3.24	6.15	7.75	3	6.48	8.19	2.25	9.72	12.79
QuIP#	4	5.83	7.25	-	-	-	3	6.29	7.82	2	9.95	11.70
SKIM	4	5.79	7.20	3.2	6.07	7.52	3	6.21	7.68	2.25	8.99	11.00
QuEPT(Uniform)	4	5.79	7.20	-	-	-	3	6.24	7.62	2	9.24	10.70
QuEPT(Mixed)	4	**5.75**	**7.15**	3.25	**5.86**	**7.25**	3	**5.98**	**7.37**	2.25	**7.76**	**8.88**

Table 8: Mixed-Precision Quantization Result of QuEPT on LLaMA-7B models.

LLaMA2-7B		4 bit			3.x bit			3 bit			2.x bit	
	Bit	Wiki	C4	Bit	Wiki	C4	Bit	Wiki	C4	Bit	Wiki	C4
FP16	16	5.47	6.97	16	5.47	6.97	16	5.47	6.97	16	5.47	6.97
SqueezeLLM	4	5.62	7.12	3.24	5.96	7.51	3	6.18	7.72	2.23	-	-
OmniQuant	4	5.74	7.35	3.25	6.03	7.75	3	6.58	8.65	2.25	11.06	15.02
QuIP#	4	5.66	7.17	-	-	-	3	6.19	7.85	2	12.30	14.80
SKIM	4	5.60	7.11	3.2	5.91	7.48	3	6.09	7.66	2.25	10.10	12.42
QuEPT(Uniform)	4	5.59	7.12	-	-	-	3	6.09	7.57	2	9.94	11.05
QuEPT(Mixed)	4	**5.54**	**7.09**	3.25	**5.65**	**7.17**	3	**5.93**	**7.37**	2.25	**8.97**	**9.07**

Table 9: Mixed-Precision Quantization Result of QuEPT on LLaMA2-7B models.

Model	**Bits**	**Method**	**PIQA**	**ARC-E**	**ARC-C**	**HellaSwag**	**WinoGrande**	**Avg.**
LLaMA2-7B	FP16	-	78.45	69.32	40.61	72.94	67.25	65.71
	W4A4KV4	SmoothQuant	51.90	26.37	25.63	27.05	48.38	35.87
		OmniQuant	66.59	53.75	31.40	55.06	54.70	52.30
		QuaRot	72.36	60.82	33.70	63.68	59.67	58.05
		QuEPT	**75.57**	**63.76**	**36.16**	**67.80**	**65.43**	**61.74**
LLaMA3-8B	FP16	-	79.71	80.09	50.51	80.09	72.84	72.65
	W4A4KV4	SmoothQuant	51.09	28.96	23.55	28.90	51.90	36.88
		OmniQuant	53.37	30.35	22.87	31.11	50.43	37.63
		QuaRot	71.98	65.67	33.76	60.68	62.15	58.85
		QuEPT	**75.52**	**75.21**	**40.95**	**70.68**	**63.14**	**65.10**

Table 10: W4-A4-KV4 quantization results of LLaMA2-7B and LLaMA3-8B models.

WikiText-2, while the best competitor, OmniQuant, scores 8.26. At higher bit-widths like W4 and W6, QuEPT consistently delivers state-of-the-art performance, often achieving the best perplexity scores or remaining highly competitive with specialized methods. This validates that our multi-bit approach is a more effective and flexible quantization solution.

Detailed Results of Mixed Precision of LLaMA-7B & LLaMA2-7B Model.

To evaluate the effectiveness of our proposed method, we conducted extensive quantization experiments on LLaMA-7B and LLaMA2-7B models. As shown in Table 8 and Table 9, we compare our method, QuEPT (in both uniform and mixed-precision variants), against several state-of-the-art Weight-Only quantization methods, including SqueezeLLM (Kim et al. 2024), OmniQuant (Shao et al. 2024), QuIP# (Tseng et al. 2024), and SKIM (Bai, Liu, and Liu 2025). The evaluation is performed across various low bit-width settings (4-bit, 3-bit, and sub-3-bit) using perplexity on the WikiText-2 and C4 datasets, where lower is better. The results clearly demonstrate that our QuEPT(Mixed) consistently outperforms all baselines across all tested bit-widths and models. Notably, the performance advantage is most significant in the ultra-low bit regimes (e.g., 2.x bit), validating the superiority of our mixed-precision strategy in

Bits	**Method**	**Criterion**	**LLaMA1-7B**		**LLaMA2-7B**		**LLaMA2-13B**		**LLaMA3-8B**	
			C4	Wiki	C4	Wiki	C4	Wiki	C4	Wiki
FP16	–	–	7.08	5.68	6.97	5.47	6.46	4.88	8.88	6.14
W2	GPTQ	Single-Bit	27.71	44.01	33.70	36.77	20.97	28.14	3.40e5	322.80
	AWQ		1.90e5	2.60e5	1.70e5	2.20e5	9.40e4	1.20e5	313.80	324.10
	OmniQuant		12.97	9.72	15.02	11.06	11.05	8.26	–	–
	QuEPT	Multi-Bit	**10.70**	**9.24**	**11.05**	**9.94**	**9.46**	**7.89**	**23.68**	**22.80**
W3	GPTQ	Single-Bit	9.49	8.06	9.81	8.37	8.02	6.44	11.63	8.27
	AWQ		13.26	11.88	23.85	24.00	13.07	10.45	11.42	8.10
	OmniQuant		8.19	6.49	8.65	6.58	7.44	5.58	–	–
	QuEPT	Multi-Bit	**7.62**	**6.24**	**7.57**	**6.09**	**6.89**	**5.30**	**10.97**	**7.77**
W4	GPTQ	Single-Bit	7.53	6.09	7.37	5.83	6.70	5.13	9.53	6.60
	AWQ		7.52	6.08	7.68	6.15	6.74	5.12	**9.41**	6.55
	OmniQuant		7.34	5.86	7.35	5.74	6.65	5.02	–	–
	SpinQuant		–	**5.76**	–	5.58	–	5.00	–	**6.49**
	Any-Precision*	Multi-Bit	–	–	7.45	5.62	–	–	–	–
	QuEPT		**7.20**	5.79	**7.12**	**5.54**	**6.56**	**4.98**	9.42	6.55
W6	GPTQ	Single-Bit	7.10	5.70	7.01	5.49	6.49	4.91	–	–
	AWQ		7.10	5.70	7.02	5.51	6.49	4.91	**8.97**	6.22
	Any-Precision*	Multi-Bit	–	–	7.27	**5.47**	–	–	–	–
	QuEPT		**7.09**	**5.68**	**7.00**	5.48	**6.48**	**4.90**	8.98	**6.22**

Table 11: Weight-only quantization results of LLaMA models. We report the perplexity on WikiText2 and C4, where Any-Precision* denotes the Any-Precision LLM method.

Model	**Calib Dataset**	**W2**		**W3**		**W4**		**W6**	
		C4	Wiki	C4	Wiki	C4	Wiki	C4	Wiki
LLaMA-7B	**Wikitext2**	10.95	7.99	7.64	6.09	7.23	5.80	7.13	5.73
	C4	10.70	9.24	7.62	6.24	7.20	5.79	7.09	5.68
LLaMA2-7B	**Wikitext2**	11.24	8.10	7.62	6.23	7.16	5.99	7.05	5.52
	C4	11.05	9.94	7.57	6.09	7.12	5.54	7.00	5.48
LLaMA2-13B	**Wikitext2**	10.41	6.91	6.91	5.21	6.60	5.00	6.53	4.94
	C4	9.46	7.89	6.89	5.30	6.56	4.98	6.48	4.90
LLaMA3-8B	**Wikitext2**	29.94	14.92	10.99	7.44	9.50	6.56	9.20	6.37
	C4	23.68	22.80	10.97	7.77	9.42	6.56	8.98	6.22

Table 12: Ablation of calibration datasets. In our main experiments, we use C4 as the calibration dataset.

preserving model performance under extreme compression.

Implementation Details

Detail Settings

In our work, all experiments on small-scale models (ViTs) were conducted on NVIDIA RTX3090 (24GB) GPUs, while the larger models (LLaMAs/LLaVA-OV) were run on NVIDIA RTXA6000 Ada (48GB) GPUs. We set the sequence length to 2048 for all LLM and LMM evaluation tasks. For all weight-activation quantization experiments, we grouped the bit-widths {4, 5, 6, 7, 8} into three sets: low {4}, mid {5, 6}, and high {7, 8}. For the weight-only quantization experiments on LLaMAs and LLaVA-OV, we grouped the bit-widths {2, 3, 4, 5, 6, 7, 8} into four sets: ex-low {2}, low {3, 4}, mid {5, 6}, and high {7, 8}. Detailed hyperparameters are in Table 14. Notably, to achieve better performance, we employ AWQ (Lin et al. 2024) for weight smoothing in our weight-only quantization of LLa-

Model	Bits	Method	WikiText2(↓)	C4(↓)	PIQA	ARC-E	ARC-C	HellaSwag	WinoGrande	0-shot[5] Avg.(↑)
LLaMA-7B	FP16	-	5.68	7.08	77.37	67.11	41.66	73.00	66.93	65.21
	W4A4	SmoothQuant	25.25	32.32	49.80	30.40	25.80	27.40	48.00	36.28
		OmniQuant	11.26	14.51	66.15	45.20	31.14	56.44	53.43	50.47
		QLLM	9.65	12.29	68.77	45.20	31.14	57.43	56.67	51.84
		QuaRot	7.08	8.73	**76.44**	50.04	**38.99**	69.39	64.72	59.92
		DuQuant	**6.40**	**7.84**	76.22	50.04	38.31	**70.09**	62.59	59.45
		QuEPT	6.45	7.92	76.00	**63.05**	36.69	68.02	**68.02**	**62.36**
	W5A5	**QuEPT**	5.91	7.33	77.64	66.08	41.13	71.59	65.82	64.45
	W6A6	SmoothQuant	6.03	7.47	76.75	51.64	39.88	71.67	65.03	60.99
		OmniQuant	5.96	7.43	77.09	51.89	40.87	71.61	65.03	61.30
		QLLM	5.89	7.34	77.26	52.02	41.04	71.40	65.19	61.38
		QuEPT	**5.80**	**7.21**	**78.18**	**66.04**	**41.47**	**72.58**	**66.69**	**64.99**
	W8A8	**QuEPT**	5.74	7.15	78.35	66.75	41.85	72.95	66.70	65.32
LLaMA2-7B	FP16	-	5.47	6.97	78.45	69.32	40.61	72.94	67.25	65.71
	W4A4	SmoothQuant	83.12	77.27	60.17	35.23	27.13	37.08	49.57	41.84
		OmniQuant	14.26	18.02	65.61	44.28	30.38	53.51	51.85	49.13
		QLLM	11.75	13.26	67.68	44.40	30.89	58.45	56.59	51.60
		QuaRot	9.66	11.98	69.48	46.25	32.76	65.69	58.56	54.55
		DuQuant	**6.28**	7.90	75.24	51.89	**36.77**	**69.54**	**62.12**	59.11
		QuEPT	6.33	**7.86**	**75.68**	**66.20**	36.26	68.49	61.48	**61.62**
	W5A5	**QuEPT**	5.66	7.17	78.07	68.65	38.82	71.53	65.04	64.42
	W6A6	SmoothQuant	6.20	7.76	75.57	53.62	39.93	71.76	66.14	61.40
		OmniQuant	5.87	7.48	76.55	53.83	40.96	55.89	65.59	58.56
		QLLM	5.91	7.31	77.48	52.99	39.33	71.38	65.98	61.43
		QuEPT	**5.53**	**7.03**	**78.45**	**68.64**	**39.76**	**72.68**	**67.88**	**65.48**
	W8A8	**QuEPT**	5.48	6.98	78.51	70.49	42.16	72.67	67.32	66.23
LLaMA2-13B	FP16	-	4.88	6.46	78.78	73.32	45.56	76.59	72.50	69.35
	W4A4	SmoothQuant	35.88	43.19	62.30	40.28	30.72	42.24	49.96	39.03
		OmniQuant	12.30	14.55	69.80	47.22	33.79	59.34	55.49	53.13
		QLLM	9.09	11.13	70.46	48.48	34.39	62.80	55.41	54.31
		DuQuant	**5.42**	**7.05**	77.31	55.60	41.55	**73.68**	66.06	62.84
		QuEPT	5.53	7.16	**77.48**	**71.25**	**41.89**	73.48	**66.14**	**66.05**
	W5A5	**QuEPT**	5.10	6.69	78.35	72.64	43.94	75.90	71.11	68.39
	W6A6	SmoothQuant	5.18	6.76	78.29	57.41	43.86	75.02	66.93	64.30
		OmniQuant	5.14	6.74	78.24	57.58	43.86	75.52	68.35	64.71
		QLLM	5.08	6.71	**78.78**	58.29	43.77	75.10	68.43	64.87
		QuEPT	**5.00**	**6.59**	78.45	**72.60**	**44.62**	**75.95**	**72.00**	**68.72**
	W8A8	**QuEPT**	4.94	6.53	78.83	72.77	44.97	76.41	72.47	69.09
LLaMA3-8B	FP16	-	6.14	8.88	79.71	80.09	50.51	80.09	72.84	72.65
	W4A4	SmoothQuant	210.19	187.93	54.57	31.90	24.23	31.26	51.14	36.97
		OmniQuant	3.64e3	2.80e3	50.22	26.94	24.57	26.55	50.20	35.70
		AffineQuant	2.12e4	3.46e4	50.71	25.93	26.02	26.07	48.46	35.44
		QuaRot	10.41	14.33	72.30	66.37	34.56	49.81	62.75	57.16
		DuQuant	8.56	**11.58**	75.68	68.48	41.81	**73.07**	66.22	65.05
		QuEPT	**8.25**	11.67	**77.26**	**74.24**	**44.28**	72.00	**67.40**	**67.04**
	W5A5	**QuEPT**	6.76	9.72	79.00	78.70	46.76	77.56	72.93	70.99
	W6A6	SmoothQuant	7.07	9.57	78.94	75.88	49.49	77.39	70.80	70.50
		OmniQuant	7.24	9.82	78.90	73.95	47.35	76.77	70.56	69.51
		AffineQuant	7.35	9.99	78.73	73.32	46.08	77.08	70.88	69.22
		QuEPT	**6.34**	**9.12**	**79.27**	**79.50**	**48.03**	**78.45**	**72.53**	**71.56**
	W8A8	**QuEPT**	6.20	8.96	79.87	80.35	50.43	80.22	73.48	72.87

Table 13: Weight-Activation quantization results of LLaMA models. We report the perplexity on WikiText2 and C4 (↓ indicates lower is better) and the accuracy on several datasets, including PIQA, ARC-E, ARC-C, HellaSwag, and WinoGrande. Avg. denotes the average accuracy across these five datasets.

MAs and LLaVA-OV, including the mixed-precision quantization detailed in the main text. Specifically, 2-bit uses group-wise quantization with a group size of 128, while other bit-widths use per-channel quantization. The methods we compare against in Table 11 are configured with these same settings. For weight and activation quantization on ViT, we use the reparameterization method provided by RepQViT (Li et al. 2023) to initialize the activation quantization parameters. To maintain consistency with the settings of PTMQ (Xu et al. 2024a), the best results are achieved when the activation quantization parameter, s_a, is fine-tuned simultaneously.

Algorithm 1: QuEPT Mixed-Precision Quantization

```
Require: A pre-trained full-precision model $\mathcal{M}_{FP16}$; A QuEPT-optimized model $\mathcal{M}_{QuEPT}$; A calibration dataset $D_{calib}$; A target average bit-width $n$; A set of candidate bit-widths $B = \{2, 3, \ldots, 8\}$
Ensure: A final mixed-precision model $\mathcal{M}_{Mixed}$
Part 1: Sensitivity Analysis
Get the list of quantizable layers $\mathcal{L} = \{l_1, l_2, \ldots, l_L\}$ from $\mathcal{M}_{QuEPT}$.
Obtain reference logits $O_{fp}$ by running inference on the full-precision model $\mathcal{M}_{FP16}$ with $D_{calib}$.
Initialize a sensitivity matrix $S$ of size $L \times |B|$.
for each candidate bit-width $b \in B$ do
    for each layer $l_i \in \mathcal{L}$ (from $i = 1$ to $L$) do
        Create a temporary model by copying the QuEPT-optimized model: $\mathcal{M}_{temp} \leftarrow \mathcal{M}_{QuEPT}$.
        Quantize only layer $l_i$ in $\mathcal{M}_{temp}$ to $b$-bit.
                ▷ All other layers $l_j$ (where $j \neq i$) retain their original precision from $\mathcal{M}_{QuEPT}$.
        Obtain quantized logits $O_q$ by running inference on $\mathcal{M}_{temp}$ with $D_{calib}$.
        Calculate the KL-divergence against the FP16 reference: $s_{i,b} \leftarrow D_{KL}(O_{fp} \,||\, O_q)$.
        Store the sensitivity: $S[i, b] \leftarrow s_{i,b}$.
    end for
end for
Part 2: Bit-width Allocation using Dynamic Programming (DP)
Define the total bit-width budget $C_{total} = n \cdot L$.
        ▷ DP Problem: Minimize total sensitivity $\sum_{i=1}^{L} S[i, b_i]$ subject to $\sum_{i=1}^{L} b_i \leq C_{total}$.
Define DP state: $DP[i][c]$ is the minimum accumulated sensitivity for the first $i$ layers using a total bit budget of $c$.
Initialize DP table: $DP[0][c] = 0$ for all $c$, and $DP[i][c] = \infty$ for $i > 0$.
Initialize a backtracking table $P$.
for $i = 1$ to $L$ do                                        ▷ Iterate through each layer
    for $c = 1$ to $C_{total}$ do                            ▷ Iterate through each possible budget
        $DP[i][c] \leftarrow \infty$
        for each candidate bit-width $b \in B$ do
            if $c \geq b$ and $DP[i-1][c-b] + S[i, b] < DP[i][c]$ then
                $DP[i][c] \leftarrow DP[i-1][c-b] + S[i, b]$
                $P[i][c] \leftarrow b$                       ▷ Record the chosen bit-width for this state
            end if
        end for
    end for
end for
Find the optimal bit-width configuration $B^* = \{b_1^*, b_2^*, \ldots, b_L^*\}$ by backtracking through table $P$.
$c_{rem} \leftarrow C_{total}$
for $i = L$ down to 1 do
    $b_i^* \leftarrow P[i][c_{rem}]$
    $c_{rem} \leftarrow c_{rem} - b_i^*$
end for
Part 3: Final Mixed-Precision Model Construction
Initialize the final model by copying the QuEPT-optimized model: $\mathcal{M}_{Mixed} \leftarrow \mathcal{M}_{QuEPT}$.
for $i = 1$ to $L$ do
    Get the optimal bit-width $b_i^*$ for layer $l_i$ from the configuration $B^*$.
    Quantize layer $l_i$ in $\mathcal{M}_{Mixed}$ to its assigned $b_i^*$-bit precision.
end for
return $\mathcal{M}_{Mixed}$
```

Model	Type	Hyperparameter	Value
ViT-S/ViT-B/DeiT-S/Swin-S	Weight-Activation	Optimizer	Adam
		Learning rate for A, B	0.001
		Learning rate for α, β	0.01
		$\lambda_1, \lambda_2, \lambda_3$	0.3,0.3,0.4
		Top p%	50%
		Epoch	20
		Calib Batchsize	32
LLaMA-7B/LLaMA2-7B/LLaMA2-13B/LLaMA3-8B	Weight-Only	Optimizer	Adam
		Learning rate for A, B	0.0007
		Learning rate for α, β	0.02
		$\lambda_1, \lambda_2, \lambda_3, \lambda_4$	0.25
		Top p%	50%
		Epoch	15
		Sequence Length	2048
		Calib Batchsize	2
LLaMA-7B/LLaMA2-7B/LLaMA2-13B	Weight-Activation	Optimizer	Adam
		Learning rate for A, B	0.00007
		Learning rate for α, β	0.2
		$\lambda_1, \lambda_2, \lambda_3$	0.3,0.3,0.4
		Top p%	50%
		Epoch	15
		Sequence Length	2048
		Calib Batchsize	2
LLaMA3-8B	Weight-Activation	Optimizer	Adam
		Learning rate for A, B	0.0001
		Learning rate for α, β	0.1
		$\lambda_1, \lambda_2, \lambda_3$	0.3,0.3,0.4
		Top p%	50%
		Epoch	15
		Sequence Length	2048
		Calib Batchsize	2
LLaVA-OV-7B	Weight-Only	Optimizer	Adam
		Learning rate for A, B	0.0007
		Learning rate for α, β	0.01
		$\lambda_1, \lambda_2, \lambda_3, \lambda_4$	0.25
		Top p%	50%
		Epoch	15
		Calib Batchsize	1
	Weight-Activation	Optimizer	Adam
		Learning rate for A, B	0.0001
		Learning rate for α, β	0.2
		$\lambda_1, \lambda_2, \lambda_3$	0.3,0.3,0.4
		Top p%	50%
		Epoch	15
		Calib Batchsize	1

Table 14: Hyperparameters of QuEPT.

References

Bai, R.; Liu, B.; and Liu, Q. 2025. SKIM: Any-bit Quantization Pushing The Limits of Post-Training Quantization. In *ICML*.

Dettmers, T.; Pagnoni, A.; Holtzman, A.; and Zettlemoyer, L. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. In *NeurIPS*.

Ding, X.; Liu, X.; Tu, Z.; Zhang, Y.; Li, W.; Hu, J.; Chen, H.; Tang, Y.; Xiong, Z.; Yin, B.; and Wang, Y. 2025. CBQ: Cross-Block Quantization for Large Language Models. In *ICLR*.

Frantar, E.; Ashkboos, S.; Hoefler, T.; and Alistarh, D. 2023. OPTQ: Accurate Quantization for Generative Pre-trained Transformers. In *ICLR*.

Kim, S.; Hooper, C.; Gholami, A.; Dong, Z.; Li, X.; Shen, S.; Mahoney, M. W.; and Keutzer, K. 2024. SqueezeLLM: Dense-and-Sparse Quantization. In *ICML*.

Li, Y.; Yu, Y.; Liang, C.; Karampatziakis, N.; He, P.; Chen, W.; and Zhao, T. 2024. LoftQ: LoRA-Fine-Tuning-aware Quantization for Large Language Models. In *ICLR*.

Li, Z.; Xiao, J.; Yang, L.; and Gu, Q. 2023. RepQ-ViT: Scale Reparameterization for Post-Training Quantization of Vision Transformers. In *ICCV*.

Lin, J.; Tang, J.; Tang, H.; Yang, S.; Dang, X.; and Han, S. 2024. AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration. In *MLSys*.

Nagel, M.; Amjad, R. A.; Van Baalen, M.; Louizos, C.; and Blankevoort, T. 2020. Up or Down? Adaptive Rounding for Post-Training Quantization. In *ICML*.

Park, Y.; Hyun, J.; Cho, S.; Sim, B.; and Lee, J. W. 2024. Any-Precision LLM: Low-Cost Deployment of Multiple, Different-Sized LLMs. In *ICML*.

Shao, W.; Chen, M.; Zhang, Z.; Xu, P.; Zhao, L.; Li, Z.; Zhang, K.; Gao, P.; Qiao, Y. J.; and Luo, P. 2024. OmniQuant: Omnidirectionally Calibrated Quantization for Large Language Models. In *ICLR*.

Tseng, A.; Chee, J.; Sun, Q.; Kuleshov, V.; and Sa, C. D. 2024. QuIP$\#$: Even Better LLM Quantization with Hadamard Incoherence and Lattice Codebooks. In *ICML*.

Xu, K.; Li, Z.; Wang, S.; and Zhang, X. 2024a. PTMQ: Post-training Multi-Bit Quantization of Neural Networks. *AAAI*.

Xu, Y.; Xie, L.; Gu, X.; Chen, X.; Chang, H.; Zhang, H.; Chen, Z.; ZHANG, X.; and Tian, Q. 2024b. QA-LoRA: Quantization-Aware Low-Rank Adaptation of Large Language Models. In *ICLR*.

Yelysei, B.; Chiaro, D.; Riccardo; and Markus, N. 2024. Low-Rank Quantization-Aware Training for LLMs. *arXiv preprint arXiv:2406.06385*.