EDGERAZOR: A LIGHTWEIGHT FRAMEWORK FOR LARGE LANGUAGE MODELS VIA MIXED-PRECISION QUANTIZATION-AWARE DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Quantization has emerged as a mainstream approach for deploying Large Language Models (LLMs) on resource-constrained devices, yet reducing precision below 4-bit typically leads to severe performance degradation or prohibitive retraining costs. In this paper, we propose EDGERAZOR, a lightweight framework for LLMs via Mixed-Precision Quantization-Aware Distillation. It contains three modules: Structural Quantization with Mixed Precision for fine-grained control over bit-widths, Layer-Adaptive Feature Distillation for dynamically selecting the most informative features for alignment, and Entropy-Aware KL Divergence for balancing forward and reverse KL to handle data from both human annotation and model distillation. With 1.88-bit weights, 8-bit activations, and 8-bit KV cache, EDGERAZOR on Qwen3-0.6B outperforms the strongest 2-bit baseline by 11.40 points in average accuracy and the strongest 3-bit baseline by 4.38 points. In terms of training efficiency, EDGERAZOR quantizes MobileLLM-350M with 4–10$\times$ fewer training tokens than the leading quantization-aware training method. On the deployment side, EDGERAZOR achieves consistently higher compression ratios than existing methods across all bit-widths. Exported to the `TQ2_0` quantization type supported by `llama.cpp`, the 1.58-bit Qwen3-0.6B quantized by EDGERAZOR reduces the model size from 1.11 GB to 0.19 GB and decodes 15.16$\times$ faster than its 16-bit counterpart. Extensive experiments on the MobileLLM and Qwen families confirm the effectiveness and efficiency of EDGERAZOR.

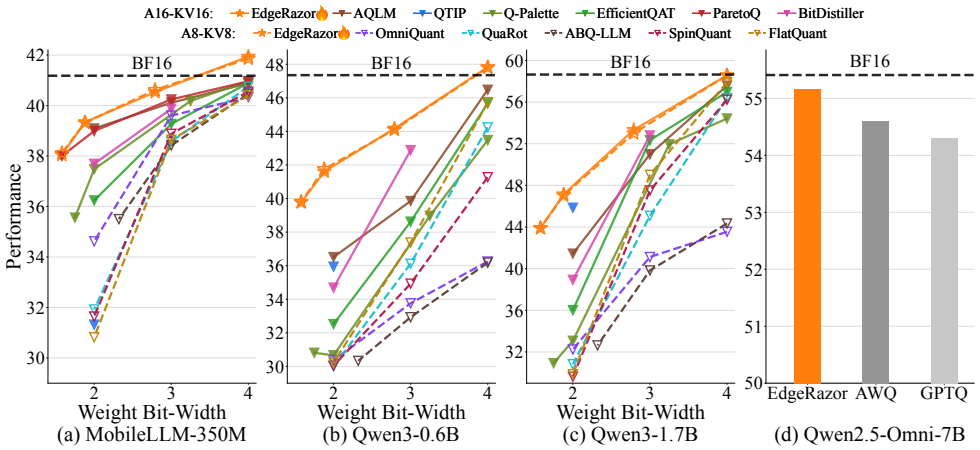

Figure 1: Performance comparison of EDGERAZOR and baselines.

1 INTRODUCTION

Large language models (LLMs) have attracted widespread attention across domains, driven by scaling laws indicating that performance improves predictably with increases in model size, dataset

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

size, and training budget. In recent years, there has been increasing demand to deploy lightweight LLMs on resource-constrained devices, where limited storage, memory, and computational capacity make it difficult for full-precision models to execute (Zheng et al., 2025). Quantization is an effective compression technique that maps full-precision weights and activations to low-bit representations (Zhang, 2026). An effective method should satisfy several prerequisites, including preserving performance, offering flexible bit-widths for deployability on resource-constrained hardware, and maintaining manageable training overhead (Tan et al., 2024).

Existing LLM quantization methods can be categorized into three paradigms: Post-Training Quantization (PTQ), Quantization-Aware Training (QAT), and Quantization-Aware Distillation (QAD). PTQ calibrates quantized parameters on a small dataset without retraining (Frantar et al., 2022; Lin et al., 2024), while incurring significant performance degradation at lower bit-widths (Dettmers & Zettlemoyer, 2023). QAT learns low-bit parameters using surrogate gradients (Bengio et al., 2013) via data-driven gradient updates to fit downstream tasks while preserving performance at below 4-bit. Nevertheless, QAT incurs substantial training costs (Liu et al., 2025a; Wang et al., 2025). QAD integrates QAT with knowledge distillation (Hinton et al., 2015; Zhou & Jiang, 2004) from a full-precision teacher to reduce the training cost (Zhang et al., 2025). However, QAD methods typically rely on heuristic approaches to pre-specify teacher layers for supervision (Xu et al., 2024a), and restrict the switching criteria between forward and reverse KL divergence (KLD) to teacher-generated data (Du et al., 2024), thereby precluding flexible data recipes that combine human-annotated and synthetic data from external models (Wu et al., 2025). Additionally, uniform and discrete bit-widths ignore parameter sensitivities (Lin et al., 2024) and restrict diverse deployment budgets (Lee & Song, 2025). Mixed-precision driven by parameter salience is confined to PTQ (Huang et al., 2025). QAT and QAD update parameters continuously, which alters salience during training (Nowak et al., 2023). A mixed-precision strategy that is robust to this alteration is therefore warranted.

In this paper, we propose EDGERAZOR, a lightweight framework for compressing LLMs with flexible bit-widths via Mixed-Precision Quantization-Aware Distillation (MPQAD). Figure 2 illustrates the workflow. EDGERAZOR comprises three configurable modules: Structural Quantization with Mixed Precision (SQMP) for fine-grained control over the matrix-wise average bit-width, Layer-Adaptive Feature Distillation (LAFD) that adaptively selects informative teacher layers for feature-level supervision, and Entropy-Aware KL Divergence (EAKLD) that balances forward and reverse KLD by the entropy of the teacher's output distribution, extending logit distillation to both human-annotated and model-generated data. These modules improve quantized models across base, instruction-tuned, and multimodal LLMs, as summarized in Figure 1. With 1.88-bit weights, 8-bit activations, and 8-bit KV cache, EDGERAZOR on Qwen3-0.6B retains promising performance on 14 downstream tasks, outperforming the state-of-the-art 2-bit baseline by 11.40 points in average accuracy and the strongest 3-bit baselines by 4.38 points. These performance gains generalize to other architectures with training efficiency, such as MobileLLM-350M, which consumes 4–10$\times$ fewer training tokens than the leading QAT method. For deployment, trained by EDGERAZOR and exported to the `TQ2_0` quantization type supported by `llama.cpp`, the 1.58-bit Qwen3-0.6B on an Apple M4 Pro chip decreases storage from 1.11 GB to 0.19 GB, while achieving a 15.16$\times$ decoding speedup over its 16-bit counterpart.

The rest of this paper is organized as follows. Section 2 reviews related works. Section 3 presents the proposed EDGERAZOR. Section 4 reports the experiments. Section 5 concludes the paper.

2 RELATED WORKS

PTQ compresses LLMs by calibrating quantized parameters on a small dataset without retraining. To maintain performance, existing methods employ local error compensation, such as weight adjustments through inverse Hessian approximation (Frantar et al., 2022), activation-aware scaling and outlier smoothing (Lin et al., 2024; Xiao et al., 2023), and vector quantization space partitioning (Tseng et al., 2024a), preserving near-lossless performance at 4-bit and above. Furthermore, mixed-precision PTQ attempts to optimize structural capacity by heuristically allocating heterogeneous bit-widths across distinct layers or groups (Guan et al., 2024; Huang et al., 2025; Lee & Song, 2025), thereby achieving better accuracy-efficiency trade-offs. Since calibration-driven strategies lack end-to-end supervision, PTQ suffers substantial performance degradation in sub-4-bit settings, thereby limiting its viability for resource-constrained scenarios (Dettmers & Zettlemoyer, 2023).

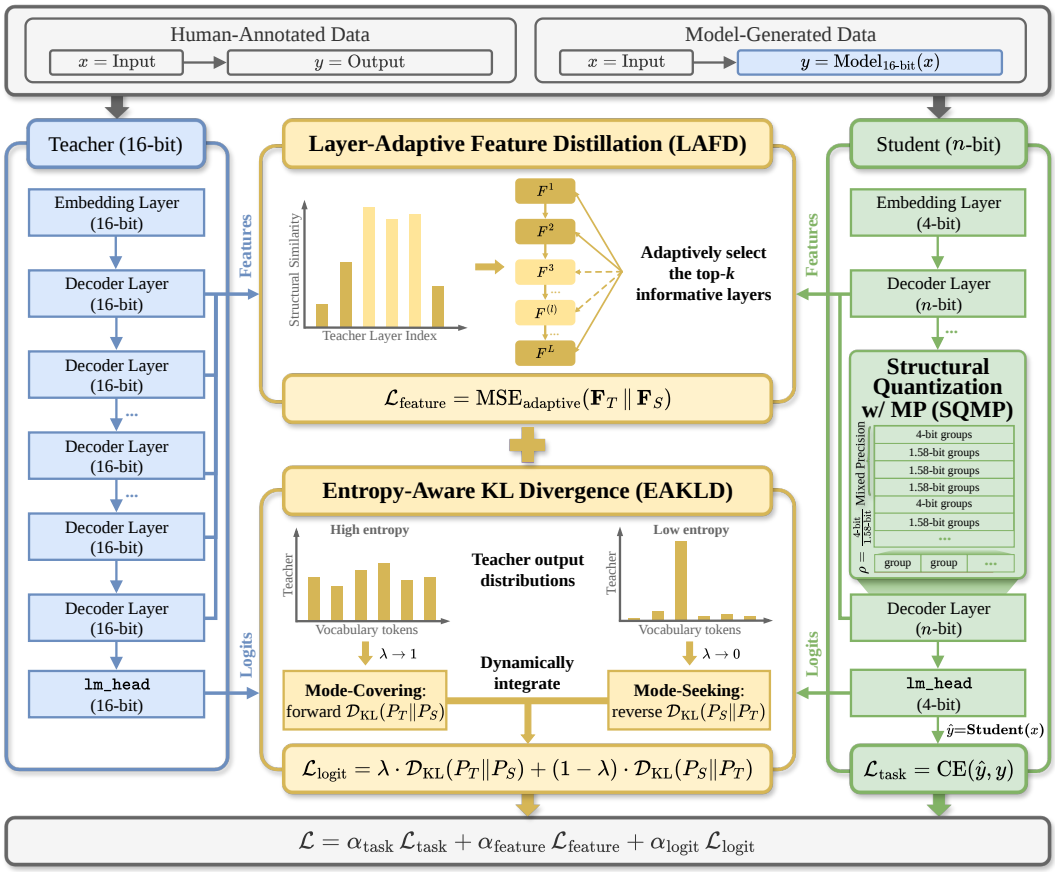

Figure 2: Workflow of the EDGERAZOR framework.

QAT uses supervised gradient updates with surrogate gradients to optimize low-bit models for target tasks (Bengio et al., 2013). Existing methods typically adopt two paradigms: training natively quantized LLMs entirely from scratch, as pioneered by BitNet (Wang et al., 2025), and fine-tuning from full-precision pre-trained models via block-wise reconstruction and optimized training budgets, exemplified by EfficientQAT (Chen et al., 2025) and ParetoQ (Liu et al., 2025a), which significantly advance the frontier of LLM compression to 2-bit and lower. Nevertheless, these methods typically require substantial computational resources and large-scale corpora to converge (Liu et al., 2025a; Wang et al., 2025), rendering the paradigm prohibitively expensive for downstream adaptations.

QAD integrates QAT with knowledge distillation (Hinton et al., 2015; Zhou & Jiang, 2004) from a full-precision teacher to alleviate the prohibitive computational demands of QAT. Existing works align output logits and intermediate features, thereby enabling 4-bit compression by data-free generation (Liu et al., 2023) and 1-bit structural decomposition (Xu et al., 2024a). Furthermore, recent advancements combine the mode-covering forward KLD (FKLD) (Hinton et al., 2015) with the mode-seeking reverse KLD, utilizing metrics such as teacher prediction confidence to improve performance at lower bit-widths (Du et al., 2024). However, existing QAD approaches are limited by heuristic layer-selection strategies that struggle to generalize across architectures and provide little guidance for selection (Wang et al., 2020), as well as by KLD-switching criteria that rely exclusively on teacher-generated data, thereby restricting the use of flexible training recipes (Wu et al., 2025).

3 EDGERAZOR

In this section, we propose EDGERAZOR. Figure 2 provides its workflow. EDGERAZOR compresses LLMs via MPQAD, which consists of three novel modules: SQMP that organizes per-channel bit-widths into a periodic super-group pattern with an adjustable ratio in Section 3.1, LAFD that aligns

intermediate representations between student and teacher models by dynamically identifying the most informative layers in Section 3.2, and EAKLD that employs the teacher's output distribution to balance forward and reverse KLD in Section 3.3. The overall training objective combines the low-bit student's task-specific cross-entropy loss $\mathcal{L}_{\text{task}}$ with the feature and logit distillation losses

$$\mathcal{L} = \alpha_{\text{task}}\,\mathcal{L}_{\text{task}} + \alpha_{\text{feature}}\,\mathcal{L}_{\text{feature}} + \alpha_{\text{logit}}\,\mathcal{L}_{\text{logit}}\,, \tag{1}$$

where $\mathcal{L}_{\text{feature}}$ and $\mathcal{L}_{\text{logit}}$ correspond to LAFD and EAKLD, respectively, and α_{task}, α_{logit}, α_{feature} are balancing coefficients for each loss value.

3.1 Structural quantization with mixed precision

Along with the per-group quantization function $Q_{n\text{-bit}}(\mathbf{X})$ in Appendix A.1, we propose SQMP to determine the average bit-width per matrix. To implement fine-grained control over bit-widths, it introduces a tunable parameter $\rho \in [0, 1]$ that indicates the proportion of weights assigned to 4-bit. We organize this structural mixed-precision quantization into a periodic super-group allocation: every $\lfloor 1/\rho \rceil$ consecutive output channels form one super-group, in which one channel is quantized to 4-bit and the remaining channels to 1.58-bit. The illustration is provided in Appendix A.2. Since every super-group maintains the same structure, tuning ρ yields fine-grained, smooth control over the fractional bit-width, accommodating diverse deployment scenarios. Thus, through super-group allocation and matrix multiplication $\mathbf{Y} = \mathbf{W}\mathbf{X}$, each output element $Y_{i,l}$ is computed as

$$Y_{i,l} = \mathbf{W}^G_{i,\cdot}\,\mathbf{X}^G_{\cdot,l} = \sum_{j=0}^{J-1} \underbrace{s^W_{i,j} \cdot s^X_{j,l}}_{①} \cdot \underbrace{Q_{n\text{-bit}}(\mathbf{W}^G_{i,j})^\top Q_{8\text{-bit}}(\mathbf{X}^G_{j,l})}_{②}\,, \quad n \in \{1.58, 4\}\,, \tag{2}$$

where $J = d_{\text{in}}/G$ denotes the total number of groups along the input dimension d_{in} for a given group size G. The group-wise vectors $\mathbf{W}^G_{i,j}$ and $\mathbf{X}^G_{j,l}$ represent the j-th group of the i-th weight output channel and the l-th activation token, respectively. The terms $s^W_{i,j}$ and $s^X_{j,l}$ are the corresponding 16-bit scaling factors, which are multiplied together to form the combined scaling factor ①, and ② is the low-bit integer dot product between the quantized weight and activation groups, where $n \in \{1.58, 4\}$ is strictly governed by the aforementioned super-group allocation. This factorization accelerates inference, as the integer arithmetic in ② can be offloaded to efficient kernels.

Unlike PTQ methods that statically preserve sensitive output channels (Huang et al., 2025), QAT and QAD update parameters during training, causing the salient weights to change accordingly (Nowak et al., 2023). We provide a theoretical justification for super-group allocation in Appendix A.3.

3.2 Layer-adaptive feature distillation

We propose LAFD to adaptively identify the most informative layers for each input using a teacher-computed metric based on structural similarity. LAFD is motivated by the observation that consecutive transformer layers do not contribute equally to the overall feature transformation (Tenney et al., 2019; Song et al., 2024). Our empirical analysis in Appendix B reveals that these transformation patterns are highly domain-dependent, indicating the limitations of heuristic or brute-force fixed-layer selection. We explicitly quantify this structural similarity using cosine similarity to assess representational transformation, as angular differences between contextual features can reflect semantic and structural changes (Ethayarajh, 2019). To quantify this transformation across layers, we compute the mean cosine similarity between the outputs of adjacent teacher layers across the set of valid token positions $\mathcal{T}$ excluding padding,

$$c_l = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \cos\Big(\mathbf{F}^{(l)}_{T,t},\ \mathbf{F}^{(l-1)}_{T,t}\Big)\,, \quad l = 1, 2, \ldots, L\,, \tag{3}$$

where $\mathbf{F}^{(l)}_{T,t} \in \mathbb{R}^d$ denotes the d-dimensional teacher features within a training batch at layer l and position t, and $\mathbf{F}^{(0)}_{T,t}$ corresponds to the output of the embedding layer. A low value of c_l indicates that layer l substantially transforms the direction of the representation. We select the k layers with the lowest scores as the targets of feature distillation,

$$\mathcal{S} = \underset{S \subseteq \{1,\ldots,L\},\ |S|=k}{\arg\min} \sum_{l \in S} c_l\,, \tag{4}$$

where $\mathcal{S}$ is the set containing selected layers. The feature distillation loss is then defined over this adaptively selected set $\mathcal{S}$ as

$$\mathcal{L}_{\text{feature}} = \text{MSE}_{\text{adaptive}}(\mathbf{F}_T \,\|\, \mathbf{F}_S) = \frac{1}{|\mathcal{S}|} \sum_{l \in \mathcal{S}} \frac{1}{|\mathcal{T}| \cdot d} \sum_{t \in \mathcal{T}} \left\| \mathbf{F}_{T,t}^{(l)} - \mathbf{F}_{S,t}^{(l)} \right\|_2^2 , \tag{5}$$

where $\mathbf{F}_{S,t}^{(l)}$ is the corresponding student features. By restricting the feature distillation loss to $\mathcal{S}$, LAFD concentrates the gradient signal on layers with the largest representational changes. This helps reduce the propagation and amplification of quantization errors through subsequent nonlinear computations. By leveraging the structural similarity metric, LAFD provides input-adaptive layer supervision while avoiding the prohibitive cost of searching over layer combinations.

3.3 Entropy-aware KL divergence

In logit distillation, the KLD is used to align the student distribution P_S with the teacher distribution P_T. The forward KLD $\mathcal{D}_{\text{KL}}(P_T \| P_S)$ is zero-avoiding, inducing mode-covering behavior, while the reverse KLD $\mathcal{D}_{\text{KL}}(P_S \| P_T)$ is zero-forcing, inducing mode-seeking behavior (Wu et al., 2025). Confidence-Aware KLD (CAKLD), introduced in BitDistiller (Du et al., 2024), balances these divergences using teacher confidence on the label tokens, but this metric leads to severe mismatch errors in both human-annotated and model-generated corpora, as demonstrated in Appendix C, thereby restricting the diversity of data recipes. To address this limitation, we propose EAKLD, which interpolates between the two objectives using a mixing coefficient λ derived solely from the teacher's output entropy. The logit distillation loss and the mixing coefficient are defined as

$$\begin{aligned} \mathcal{L}_{\text{logit}} = \mathcal{D}_{\text{EAKLD}}(P_T \,\|\, P_S) &= \lambda \underbrace{\mathcal{D}_{\text{KL}}(P_T \,\|\, P_S)}_{\text{forward KLD}} + (1-\lambda) \underbrace{\mathcal{D}_{\text{KL}}(P_S \,\|\, P_T)}_{\text{reverse KLD}} , \\ \text{with} \quad \lambda &= \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\frac{1}{|y|} \sum_{i=1}^{|y|} \frac{\min\Big(H\big(P_T(y_i|x, y_{<i})\big),\ \log K \Big)}{\log K} \right] , \end{aligned} \tag{6}$$

where $\mathbb{D}$ is the data within a training batch, $|y|$ is the number of tokens in the response sequence, and $H(P_T(x, y_{<i}))$ denotes the entropy of the teacher's predictive distribution at position i conditioned on the input x and the preceding tokens $y_{<i}$. The entropy is formulated as

$$H\big(P_T(y_i|x, y_{<i})\big) = -\sum_{v \in \mathcal{V}} P_T(v \mid x, y_{<i}) \log P_T(v \mid x, y_{<i}) , \tag{7}$$

where $\mathcal{V}$ is the vocabulary set, and the denominator $\log K$ represents the maximum entropy of an uniform distribution. When the teacher disperses probability evenly among candidates, the entropy increases, causing λ to rise and adaptively strengthening the forward KLD, thereby encouraging mode-covering. Conversely, when the teacher places high confidence in candidates, λ decays, thereby prioritizing the reverse KLD for precise mode-seeking. Furthermore, tuning the hyper-parameter K deterministically alters the upper-bound entropy, serving as the primary control mechanism to adjust the entire dataset's aggregate tendency toward either divergence strategy.

By adapting logit distillation to use entropy exclusively, EAKLD captures the shape of the teacher's output distribution and avoids the need for teacher-generated labels, thereby supporting training data that comprise human-annotated data and data generated by other high-quality models.

4 Experiments

In this section, we conduct comprehensive experiments to validate the effectiveness and efficiency of our proposed framework along with its three modules.

4.1 Experimental setup

Models. We apply EdgeRazor to four models spanning three categories: MobileLLM-350M (base) (Liu et al., 2025a), Qwen3-0.6B and Qwen3-1.7B (instruction-tuned) (Yang et al., 2025), and Qwen2.5-Omni-7B (multimodal) (Xu et al., 2025).

Table 1: Hyperparameters for EDGERAZOR.

Models	ρ	Bit-widths	LRs	Training	α_{task}	α_{feature}	α_{logit}
MobileLLM-350M	1	4.00	2×10^{-5}	2 epochs	0.50	0.10	2.0
	$1/2$	2.79		4 epochs	0.50	0.10	2.0
	$1/8$	1.88		5 epochs	0.20	1.00	4.0
	0	1.58		5 epochs	0.20	1.00	4.0
Qwen3-0.6B	1	4.00	2×10^{-5}	2k steps	0.05	0.50	2.0
	$1/2$	2.79		1 epoch	0.10	0.10	2.0
	$1/8$	1.88		1 epoch	0.10	0.10	2.0
	0	1.58		1 epoch	0.10	0.10	2.0
Qwen3-1.7B	1	4.00	2×10^{-5}	2k steps	0.05	0.50	2.0
	$1/2$	2.79		2 epochs	0.10	0.10	2.0
	$1/8$	1.88		2 epochs	0.10	0.10	2.0
	0	1.58		2 epochs	0.10	0.10	2.0
Qwen2.5-Omni-7B	1	4.00	5×10^{-6}	2 epochs	0.10	0.20	2.0

Datasets. We train the base and instruction-tuned LLMs on a mixture of human-annotated and model-generated corpora (Li et al., 2025a; Zhao et al., 2025), combined with the training splits of downstream tasks (Bisk et al., 2020; Clark et al., 2019; 2018; Mihaylov et al., 2018; Sakaguchi et al., 2021; Sap et al., 2019; Zellers et al., 2019). All synthetic samples are generated by high-quality LLMs rather than their corresponding teachers. In addition, the multimodal LLM is trained on 10K TGIF (Li et al., 2016) clips re-encoded at 30 FPS, with video-understanding responses distilled from the teacher. Appendix D.1 details the data mixtures.

Hyperparameters. All low-bit LLMs are trained on 8×NVIDIA A100-80 GB GPUs. We configure the sequence length to 1024 and the group sizes to 64, 128, and 256 for MobileLLM, Qwen2.5, and Qwen3, respectively. The embedding and `lm_head` layers are compressed to 4-bit. We select the best hyperparameters using 1024 validation samples from WikiText2 (Merity et al., 2016). For training, We adopt the AdamW optimizer and set β=2.0, ϵ=10^{-5}, k=3 in LAFD, and K=16 in EAKLD. For scheduling, Qwen3-0.6B and 1.7B use a constant learning rate with a 0.05 warmup ratio and a batch size of 1024. MobileLLM-350M and Qwen2.5-Omni-7B utilize a cosine schedule with a 0.01 warmup ratio and batch sizes of 1920 and 64. Additional details are provided in Table 1.

Baselines. We compare EDGERAZOR against weight-only and weight-activation baselines on both effectiveness and efficiency. PTQ baselines are AQLM (Egiazarian et al., 2024), QTIP (Tseng et al., 2024b), Q-Palette (Lee & Song, 2025), OmniQuant (Shao et al., 2024), QuaRot (Ashkboos et al., 2024), ABQ-LLM (Zeng et al., 2025), SpinQuant (Liu et al., 2025b), and FlatQuant (Sun et al., 2025). QAT baselines include EfficientQAT (Chen et al., 2025) and ParetoQ (Liu et al., 2025a). QAD baseline with the same training datasets is BitDistiller (Du et al., 2024). The baseline group sizes are 64, 128, and 256 for MobileLLM, Qwen2.5, and Qwen3. We also provide per-task evaluation results against additional baselines in Appendix D.3. For the ablation studies, we compare three aspects: SQMP with super-group allocation (Figure 6) against other precision formats; LAFD against fixed-layer feature distillation; and EAKLD against forward KLD.

Evaluation. Table 8 summarizes the evaluation protocols. Since perplexity is a weak proxy for downstream behaviour (Xu et al., 2024b), we therefore prioritize downstream benchmarks over perplexity, so that the reported scores reflect reasoning and generation ability. The text LLMs are benchmarked across tasks with random seed 42, including commonsense reasoning (Bisk et al., 2020; Clark et al., 2019; 2018; Sakaguchi et al., 2021; Sap et al., 2019; Zellers et al., 2019), reading comprehension (Mihaylov et al., 2018), trustworthiness (Hendrycks et al., 2020a), truthfulness (Lin et al., 2022), and knowledge (Hendrycks et al., 2020b), using the `lm_eval` v0.4.9.1. The instruction-tuned LLMs are evaluated on additional tasks: instruction following (Zhou et al., 2023), mathematics (Cobbe et al., 2021), and coding (Chen et al., 2021). Qwen2.5-Omni-7B is evaluated on two video understanding tasks (Fu et al., 2025; Zhou et al., 2025) using the `lmms_eval` v0.5.

Table 2: Performance of EDGERAZOR and baselines on MobileLLM-350M, Qwen3-0.6B, and Qwen3-1.7B. **Bold** denotes the best result and underline denotes the second best result. H-Average refers to held-out benchmarks in average, including MMLU, IFEval, GSM8K, and HumanEval.

W-A-KV	Methods	MobileLLM-350M	Qwen3-0.6B		Qwen3-1.7B	
		Average (↑)	Average (↑)	H-Average (↑)	Average (↑)	H-Average (↑)
16-16-16	BF16	48.67	47.35	44.32	58.64	64.61
4-16-16	AQLM	48.31	46.48	**42.41**	57.57	**62.69**
4-16-16	Q-Palette	48.42	43.48	35.12	54.42	54.33
4-16-16	EfficientQAT	48.32	45.67	38.84	56.96	58.90
4-16-16	EDGERAZOR	**49.52**	**47.83**	42.24	**58.56**	59.84
4-8-8	QuaRot	48.16	44.24	33.91	56.38	60.90
4-8-8	SpinQuant	47.92	41.27	27.40	56.22	59.02
4-8-8	FlatQuant	47.73	45.74	37.26	57.90	**62.67**
4-8-8	EDGERAZOR	**49.42**	**47.80**	**42.48**	**58.57**	60.56
3-16-16	EfficientQAT	46.41	38.62	21.83	52.34	48.72
3-16-16	ParetoQ	47.56	–	–	–	–
3-16-16	BitDistiller	47.12	42.88	29.95	52.79	48.66
2.79-16-16	EDGERAZOR	**47.83**	**44.17**	**36.51**	**53.33**	**52.23**
3-8-8	OmniQuant	46.80	33.76	10.12	41.12	15.03
3-8-8	SpinQuant	45.94	34.93	15.06	47.51	33.95
3-8-8	FlatQuant	45.66	37.38	18.85	49.01	39.82
2.79-8-8	EDGERAZOR	**47.92**	**44.10**	**36.32**	**53.00**	**51.23**
2-16-16	AQLM	46.19	36.51	19.26	41.44	21.92
2-16-16	QTIP	37.01	35.94	12.48	45.85	34.36
2-16-16	BitDistiller	44.54	34.69	13.47	38.92	16.60
1.88-16-16	EDGERAZOR	**46.42**	**41.60**	**29.89**	**47.14**	**39.69**
2-8-8	OmniQuant	40.91	30.36	9.19	32.24	8.60
2-8-8	ABQ-LLM	41.96	30.33	8.68	32.63	8.60
2-8-8	FlatQuant	36.43	30.23	8.56	29.87	8.82
1.88-8-8	EDGERAZOR	**46.42**	**41.76**	**30.59**	**47.03**	**39.84**
1.75-16-16	Q-Palette	42.02	30.81	9.02	30.92	8.91
1.58-16-16	ParetoQ	44.90	–	–	–	–
1.58-16-16	EDGERAZOR	**44.91**	**39.77**	**26.90**	**43.89**	**32.69**
1.58-8-8	EDGERAZOR	**44.97**	**39.81**	**27.28**	**43.91**	**32.33**

Table 3: Performance of EDGERAZOR and baselines on Qwen2.5-Omni-7B. **Bold** denotes the best result and underline denotes the second best result.

W-A-KV	Methods	Video-MME	MLVU	Average (↑)
16-16-16	BF16	62.81	48.01	55.41
4-16-16	AWQ	61.78	47.40	54.59
4-16-16	GPTQ	60.51	**48.06**	54.29
4-16-16	EDGERAZOR	**62.44**	47.85	**55.15**

To evaluate deployment efficiency, we benchmark on an Apple M4 Pro chip using `llama-bench` in `llama.cpp` b6300 under 100 repetitions. Additional results are provided in Appendix D.6.

4.2 MAIN RESULTS

Table 2 reports the average performance and H-Average, while Appendix D.3 provides per-task results and Appendix E presents the generation example. Average is the mean across all tasks, and H-Average is the mean across four held-out benchmarks, MMLU, IFEval, GSM8K, and HumanEval, whose training splits are absent from the mixed training data. We omit H-Average for MobileLLM-350M (base), which is not instruction-tuned and scores 0.00 on GSM8K and HumanEval at BF16.

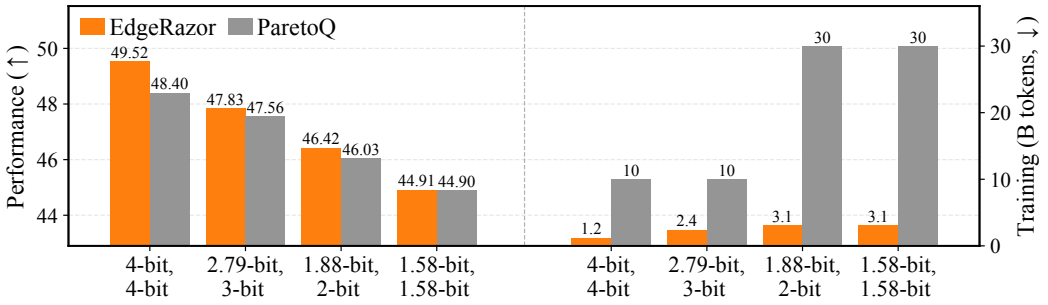

Figure 3: Performance and training budgets of EDGERAZOR and ParetoQ on MobileLLM-350M. The training budgets are reported in tokens consumed during training.

It is observed that EDGERAZOR attains the best average in every setting, with a lead that widens as the bit-width decreases. On Qwen3-0.6B, the lead over the strongest baseline grows from 2.06 at W4-A8-KV8 to 11.40 at W2-A8-KV8. At W1.88-A16-KV16, it reaches 41.60, exceeding the PTQ baseline AQLM by 5.09 and the QAD baseline BitDistiller by 6.91 when both are trained on the same datasets. At around 1.58-bit, it matches the QAT baseline ParetoQ on MobileLLM-350M and leads the PTQ baseline Q-Palette by 8.96 and 12.97 on the two Qwen3 models. The lead is even larger on H-Average: at 3-bit and below, every PTQ baseline stays below 10, and the margins over the strongest baseline exceed those on average for both Qwen3 models. Since the training splits of these four benchmarks are excluded from training, the advantage of EDGERAZOR is not an artifact of overlap between training and evaluation. Furthermore, EDGERAZOR is robust to both weight-activation quantization and aggressive low bit-widths. Across all settings, the weight-activation and weight-only variants differ by at most 0.33 points on average and 1.00 point on H-Average.

Table 3 presents the performance of EDGERAZOR and weight-only baselines on Qwen2.5-Omni-7B. It indicates that the advantage carries over to the multimodal LLM. With the vision encoder, embedding, and `lm_head` quantized as well, EDGERAZOR achieves 55.15 on Qwen2.5-Omni-7B at W4-A16-KV16, against 54.59 for AWQ and 54.29 for GPTQ. Overall, one exception is the 4-bit regime, where AQLM and FlatQuant outperform EDGERAZOR on H-Average. At 3-bit and below, EDGERAZOR achieves the best average and H-Average across all settings. Beyond accuracy, EDGERAZOR also improves training efficiency. On MobileLLM-350M, it uses 4–10× fewer tokens than the QAT baseline ParetoQ while achieving higher performance, as shown in Figure 3.

4.3 ABLATION STUDIES

Table 4: Configurations of the ablation studies.

Methods	Bit-width Allocation			Feature Distillation		Logit Distillation	
	Super-group	Stacked	Uniform	LAFD	Fixed	EAKLD	FKLD
Baseline	✓			✓		✓	
Ablation 1		✓		✓		✓	
Ablation 2			✓	✓		✓	
Ablation 3	✓				✓	✓	
Ablation 4	✓			✓			✓
Ablation 5			✓		✓		✓

Tables 4 and 5 report the ablation configurations and results on Qwen3-0.6B: EDGERAZOR serves as the baseline, Ablations 1 to 5 remove or replace one or more modules, and uniform allocation assigns all weights to 2-bit precision. Per-task results are provided in Appendix D.5. Super-group allocation outperforms the stacked variant by 1.00 and 1.32, and the uniform 2-bit variant by 0.62 and 0.81. Removing LAFD results in decreases of 1.20 and 1.52, whereas removing EAKLD leads to decreases of 1.46 and 1.88, demonstrating the effectiveness of both modules. Standard QAD (Ablation 5) underperforms by 2.58 and 3.01, exceeding the impact of any individual module removal and indicating that the improvement arises from the combination of all three proposed modules.

Table 5: Performance of ablation studies. Δ_{Baseline} denotes the difference relative to the baseline.

W-A-KV	Methods	Average (↑)	Δ_{Baseline}	W-A-KV	Methods	Average (↑)	Δ_{Baseline}
1.88-16-16	Baseline	41.60	–	1.88-8-8	Baseline	41.76	–
1.88-16-16	Ablation 1	40.60	-1.00	1.88-16-16	Ablation 1	40.44	-1.32
2-16-16	Ablation 2	40.98	-0.62	2-8-8	Ablation 2	40.95	-0.81
1.88-16-16	Ablation 3	40.40	-1.20	1.88-8-8	Ablation 3	40.24	-1.52
1.88-16-16	Ablation 4	40.14	-1.46	1.88-8-8	Ablation 4	39.88	-1.88
2-16-16	Ablation 5	39.02	-2.58	2-8-8	Ablation 5	38.75	-3.01

4.4 Compression and Acceleration

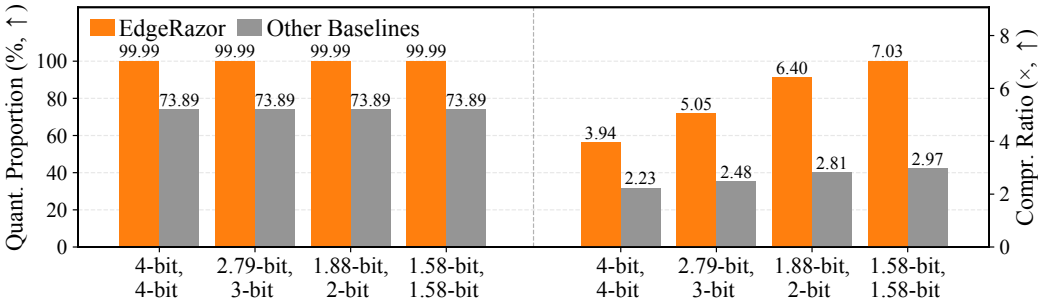

Figure 4: Efficiency comparison of EdgeRazor and other baselines at each bit-width.

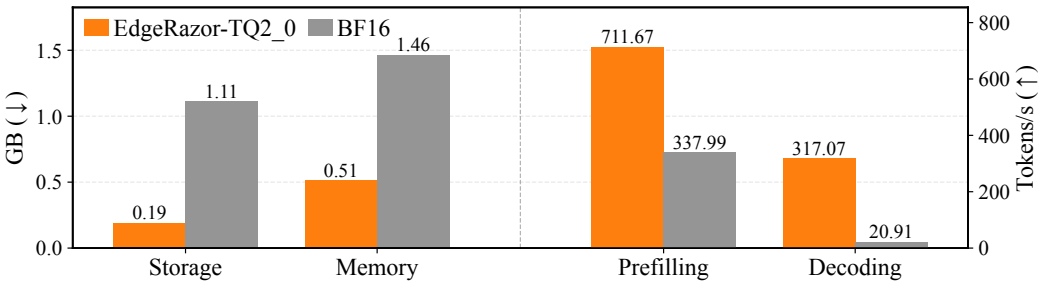

Figure 5: Efficiency comparison of deploying 1.58-bit and 4-bit Qwen3-0.6B via `llama.cpp`.

Beyond effectiveness, deployment efficiency matters for edge applications. Figure 4 compares two efficiency metrics on Qwen3-0.6B. It reveals that quantizing the embedding and `lm_head` alongside the decoder layers yields consistently higher quantization proportions and compression ratios than other baselines across all bit-widths. In Figure 5, 1.58-bit EdgeRazor under the `TQ2_0` quantization type reduces storage from 1.11 GB to 0.19 GB and memory from 1.46 GB to 0.51 GB, while accelerating prefilling and decoding by 2.11× and 15.16× over BF16. Compression thus translates into on-device speedups, and Appendix D.6 reports results on additional LLMs and chips.

5 Conclusions

In this paper, we proposed EdgeRazor, a lightweight framework for compressing LLMs into flexible bit-widths via MPQAD. It consists of three novel modules: SQMP, LAFD, and EAKLD. Extensive evaluations across base, instruction-tuned, and multimodal LLMs demonstrated that EdgeRazor outperforms state-of-the-art baselines at most bit-widths in terms of downstream performance and training efficiency. Practical benchmarks on `llama.cpp` demonstrated that EdgeRazor transforms aggressive quantization from a fragile trade-off into a resilient approach for deploying lightweight LLMs on resource-constrained hardware.

AI USE STATEMENT

In this work, we used generative AI tools for manuscript translation and polishing. We have not used generative AI tools for the following tasks, including generating synthetic data sets, developing theoretical models or conceptual frameworks, designing or providing feedback on research methodology or experiments, cleaning and reformatting datasets, and interpreting results. The following tasks do not apply to this work: Formulate mathematical claims; provide critical ingredients for proving mathematical claims; assist in the writing of proofs; propose or refine hypotheses; support qualitative and thematic data analysis. We have reviewed all AI-assisted work; even the AI-assisted translation was checked by the authors to ensure that it preserved the intended meaning. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-free 4-bit inference in rotated LLMs. In *Advances in Neural Information Processing Systems 37*, pp. 100213–100240, 2024.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pp. 7432–7439, 2020.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. EfficientQAT: Efficient quantization-aware training for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 10081–10100, 2025.

Wenhua Cheng, Weiwei Zhang, Haihao Shen, Yiyang Cai, Xin He, Lv Kaokao, and Yi Liu. Optimize weight rounding via signed gradient descent for the quantization of LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 11332–11350, 2024.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 2924–2936, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Tim Dettmers and Luke Zettlemoyer. The case for 4-bit precision: K-bit inference scaling laws. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 7750–7774, 2023.

Dayou Du, Yijia Zhang, Shijie Cao, Jiaqi Guo, Ting Cao, Xiaowen Chu, and Ningyi Xu. BitDistiller: Unleashing the potential of sub-4-bit LLMs via self-distillation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 102–116, 2024.

Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. Extreme compression of large language models via additive quantization. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 12284–12303, 2024.

Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 55–65, 2019.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *Proceedings of the 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24108–24118, 2025.

Ziyi Guan, Hantao Huang, Yupeng Su, Hong Huang, Ngai Wong, and Hao Yu. APTQ: Attention-aware post-training mixed-precision quantization for large language models. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*, pp. 1–6, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. Aligning AI with shared human values. *arXiv preprint arXiv:2008.02275*, 2020a.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020b.

Jung Hwan Heo, Jeonghoon Kim, Beomseok Kwon, Byeongwook Kim, Se Jung Kwon, and Dongsoo Lee. Rethinking channel dimensions to isolate outliers for low-bit weight quantization of large language models. In *Proceedings of the 12th International Conference on Learning Representations*, pp. 12744–12762, 2024.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Wei Huang, Yangdong Liu, Haotong Qin, Ying Li, Shiming Zhang, Xianglong Liu, Michele Magno, and Xiaojuan Qi. BiLLM: Pushing the limit of post-training quantization for LLMs. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 20023–20042, 2024.

Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and Xiaojuan Qi. SliM-LLM: Salience-driven mixed-precision quantization for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 25672–25692, 2025.

Deokjae Lee and Hyun Oh Song. Q-Palette: Fractional-bit quantizers toward optimal bit allocation for efficient LLM deployment. *arXiv preprint arXiv:2509.20214*, 2025.

Jijie Li, Li Du, Hanyu Zhao, Bowen Zhang, Liangdong Wang, Boyan Gao, Guang Liu, and Yonghua Lin. Infinity Instruct: Scaling instruction selection and synthesis to enhance language models. *arXiv preprint arXiv:2506.11116*, 2025a.

Yuhang Li, Ruokai Yin, Donghyun Lee, Shiting Xiao, and Priyadarshini Panda. GPTAQ: Efficient finetuning-free quantization for asymmetric calibration. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 36690–36706, 2025b.

Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. TGIF: A new dataset and benchmark on animated gif description. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4641–4650, 2016.

Zhiteng Li, Xianglong Yan, Tianao Zhang, Haotong Qin, Dong Xie, Jiang Tian, Zhongchao Shi, Linghe Kong, Yulun Zhang, and Xiaokang Yang. ARB-LLM: Alternating refined binarizations for large language models. In *Proceedings of the 13th International Conference on Learning Representations*, pp. 93900–93912, 2025c.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of the 6th Conference on Machine Learning and Systems*, volume 6, pp. 87–100, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 3214–3252, 2022.

Yifei Liu, Jicheng Wen, Yang Wang, Shengyu Ye, Li Lyna Zhang, Ting Cao, Cheng Li, and Mao Yang. VPTQ: Extreme low-bit vector post-training quantization for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8181–8196, 2024.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. LLM-QAT: Data-free quantization aware training for large language models. *arXiv preprint arXiv:2305.17888*, 2023.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. ParetoQ: Scaling laws in extremely low-bit LLM quantization. *arXiv preprint arXiv:2502.02631*, 2025a.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. SpinQuant: LLM quantization with learned rotations. In *Proceedings of the 13th International Conference on Learning Representations*, pp. 92009–92032, 2025b.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, 2018.

Aleksandra Nowak, Bram Grooten, Decebal Constantin Mocanu, and Jacek Tabor. Fantastic weights and how to find them: Where to prune in dynamic sparse training. In *Advances in Neural Information Processing Systems 36*, pp. 55160–55192, 2023.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 4463–4473, 2019.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. OmniQuant: Omnidirectionally calibrated quantization for large language models. In *Proceedings of the 12th International Conference on Learning Representations*, pp. 45472–45496, 2024.

Jiwon Song, Kyungseok Oh, Taesu Kim, Hyungjun Kim, Yulhwa Kim, and Jae-Joon Kim. SLEB: Streamlining LLMs through redundancy verification and elimination of transformer blocks. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 46136–46155, 2024.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, Xin Jiang, Wulong Liu, and Jun Yao. FlatQuant: Flatness matters for LLM quantization. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 57587–57613, 2025.

Fuwen Tan, Royson Lee, Łukasz Dudziak, Shell Xu Hu, Sourav Bhattacharya, Timothy Hospedales, Georgios Tzimiropoulos, and Brais Martinez. MobileQuant: Mobile-friendly quantization for on-device language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 9761–9771, 2024.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, 2019.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP#: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 48630–48656, 2024a.

Albert Tseng, Qingyao Sun, David Hou, and Christopher M De Sa. QTIP: Quantization with trellises and incoherence processing. In *Advances in Neural Information Processing Systems 37*, pp. 59597–59620, 2024b.

Hongyu Wang, Shuming Ma, Lingxiao Ma, Lei Wang, Wenhui Wang, Li Dong, Shaohan Huang, Huaijie Wang, Jilong Xue, Ruiping Wang, Jihao Bao, Conghui He, and Furu Wei. BitNet: 1-bit pre-training for large language models. *Journal of Machine Learning Research*, 26(125):1–29, 2025.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems 33*, pp. 5776–5788, 2020.

Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. Rethinking kullback-leibler divergence in knowledge distillation for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5737–5755, 2025.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 38087–38099, 2023.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, et al. Qwen2.5-omni technical report. *arXiv preprint arXiv:2503.20215*, 2025.

Yuzhuang Xu, Xu Han, Zonghan Yang, Shuo Wang, Qingfu Zhu, Zhiyuan Liu, Weidong Liu, and Wanxiang Che. OneBit: Towards extremely low-bit large language models. In *Advances in Neural Information Processing Systems 37*, pp. 66357–66382, 2024a.

Zhichao Xu, Ashim Gupta, Tao Li, Oliver Bentham, and Vivek Srikumar. Beyond perplexity: Multi-dimensional safety evaluation of LLM compression. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15359–15396, 2024b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MAmmoTH: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

Chao Zeng, Songwei Liu, Yusheng Xie, Hong Liu, Xiaojian Wang, Miao Wei, Shu Yang, Fangmin Chen, and Xing Mei. ABQ-LLM: Arbitrary-bit quantized inference acceleration for large language models. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence*, pp. 22299–22307, 2025.

Cheng Zhang, Jianyi Cheng, George A Constantinides, and Yiren Zhao. LQER: Low-rank quantization error reconstruction for LLMs. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 58763–58779, 2024.

Shao-Qun Zhang. On the expressive power of weight quantization in large language models. *arXiv preprint arXiv:2606.22249*, 2026.

Shu-Hao Zhang, Wei-Cheng Tang, Chen Wu, Peng Hu, Nan Li, Liang-Jie Zhang, Qi Zhang, and Shao-Qun Zhang. TernaryCLIP: Efficiently compressing vision-language models with ternary weights and distilled knowledge. *arXiv preprint arXiv:2510.21879*, 2025.

Han Zhao, Haotian Wang, Yiping Peng, Sitong Zhao, Xiaoyu Tian, Shuaiting Chen, Yunjie Ji, and Xiangang Li. 1.4 million open-source distilled reasoning dataset to empower large language model training. *arXiv preprint arXiv:2503.19633*, 2025.

Yue Zheng, Yuhao Chen, Bin Qian, Xiufang Shi, Yuanchao Shu, and Jiming Chen. A review on edge large language models: Design, execution, and applications. *ACM Computing Surveys*, 57 (8):1–35, 2025.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. MLVU: Benchmarking multi-task long video understanding. In *Proceedings of the 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13691–13701, 2025.

Zhi-Hua Zhou and Yuan Jiang. Nec4. 5: Neural ensemble based c4. 5. *IEEE Transactions on Knowledge and Data Engineering*, 16(6):770–773, 2004.

A MIXED-PRECISION QUANTIZATION

A.1 QUANTIZATION FUNCTION FOR WEIGHTS AND ACTIVATIONS

In this section, we employ the per-group symmetric quantization for both weights and activations in our framework. Let $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ denote a weight matrix and $\mathbf{X} \in \mathbb{R}^{d_{\text{in}} \times L}$ denote the corresponding activation matrix, where L is the sequence length. Given a group size G, we partition $\mathbf{W}$ and $\mathbf{X}$ along the input dimension into $J = d_{\text{in}}/G$ groups per output channel to obtain $\mathbf{W}^G \in \mathbb{R}^{d_{\text{out}} \times J \times G}$ and $\mathbf{X}^G \in \mathbb{R}^{J \times G \times L}$. The j-th group of the i-th output channel is defined as $\mathbf{W}^G_{i,j} = \mathbf{W}[i,\ jG : (j{+}1)G] \in \mathbb{R}^G$, with $W^G_{i,j,k}$ denoting its k-th element. Similarly, the j-th input-channel group for token l is defined as $\mathbf{X}^G_{j,l} = \mathbf{X}[jG : (j{+}1)G,\ l] \in \mathbb{R}^G$. Each group is independently quantized to n-bit through a quantization function applicable to both weights and activations,

$$Q_{n\text{-bit}}(\mathbf{W}^G_{i,j}) = \begin{cases} \text{clip}\left(\left\lfloor \dfrac{\mathbf{W}^G_{i,j}}{s_{i,j}} \right\rceil,\ -1,\ 1\right) \text{ with } s_{i,j} = \max\left(\dfrac{\beta}{G}\displaystyle\sum_k |W^G_{i,j,k}|,\ \epsilon\right), & \text{if } n = 1.58 \\ \left\lfloor \dfrac{\mathbf{W}^G_{i,j}}{s_{i,j}} \right\rceil \text{ with } s_{i,j} = \max\left(\dfrac{\max_k |W^G_{i,j,k}|}{2^{n-1}-1},\ \epsilon\right), & \text{if } n \in \{4,\ 8\} \end{cases} \tag{8}$$

where $\lfloor \cdot \rceil$ denotes rounding to the nearest integer, $s_{i,j}$ is the scaling factor, β is a tunable scaling coefficient for ternarization, and ϵ is a small constant that prevents division by zero. The 1.58-bit branch, as $\log_2 3 \approx 1.58$, quantizes each weight to the ternary set $\{-1, 0, +1\}$, with the scaling factor derived from the group mean absolute value. The n-bit branch, where $n \in \{4, 8\}$, quantizes weights to the symmetric integer range such as $[-7,\ 7]$ for 4-bit and $[-127,\ 127]$ for 8-bit, with the scaling factor determined by the group-wise maximum absolute value. Specifically, the activation groups are uniformly quantized to 8-bit under weight-activation quantization or retained at 16-bit under weight-only quantization.

A.2 STRATEGIES OF STRUCTURAL QUANTIZATION WITH MIXED PRECISION

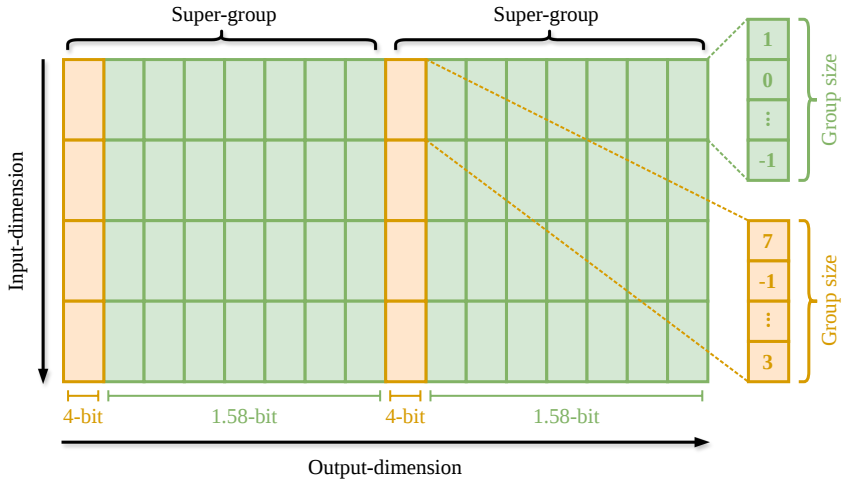

Figure 6: Super-group allocation for weight matrices, visualized via the transposed matrix $\mathbf{W}^T$.

In this section, we present strategies for allocating high-precision and low-precision groups within weight matrices in our proposed Structural Quantization with Mixed Precision (SQMP) module. Figure 7 shows the stacked allocation, while Figure 6 depicts the super-group allocation. For instance, setting $\rho = 1/8$ places one 4-bit output channel followed by seven 1.58-bit output channels within each super-group, resulting in an effective bit-width of roughly 1.88-bit.

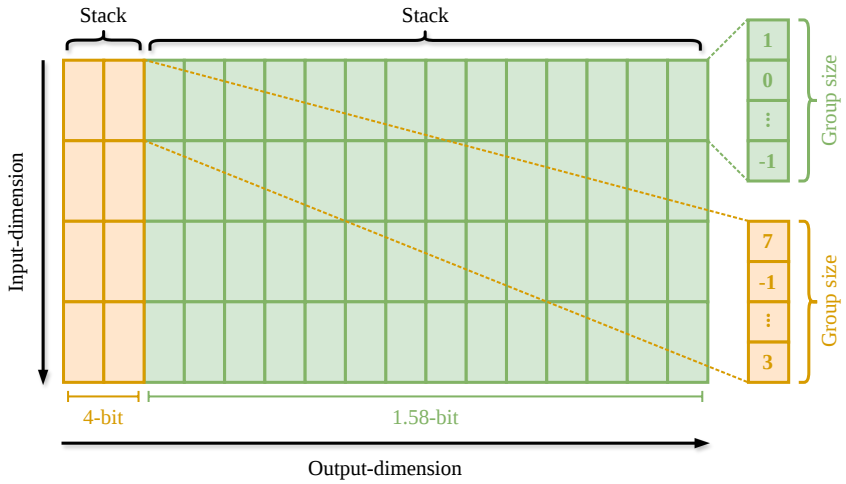

Figure 7: Stacked allocation for weight matrices, visualized via the transposed matrix $\mathbf{W}^T$.

A.3 Proof of super-group allocation in SQMP

In this section, we justify the super-group allocation by showing that, under a bounded-variation model of salience alteration during training, it minimizes a discrepancy-based worst-case upper bound on the salience-alignment error among the allocation schemes considered in this paper.

Definition. Let $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ and $\mathbf{X} \in \mathbb{R}^{d_{\text{in}} \times L}$ denote the full-precision weights and activations, and let

$$\Delta \mathbf{Y} \;=\; \mathbf{W}\mathbf{X} - Q(\mathbf{W})\mathbf{X} \tag{9}$$

denote the quantization error. For each output row i, let $a_i \in \{0, 1\}$ indicate the precision assignment, where $a_i = 1$ denotes a 4-bit assignment and $a_i = 0$ denotes a 1.58-bit assignment. Under the bit-width budget and specific proportion,

$$\frac{1}{d_{\text{out}}} \sum_{i=1}^{d_{\text{out}}} a_i \;=\; \rho, \;\; \text{and} \;\; N \;:=\; \sum_{i=1}^{d_{\text{out}}} a_i \;=\; \rho\, d_{\text{out}}, \tag{10}$$

we adopt the row-separable surrogate

$$\mathcal{L}(\mathbf{a}) \;\lesssim\; \sum_{i=1}^{d_{\text{out}}} \big((1 - a_i)\, e_L + a_i\, e_H\big)\, S_i \;=\; \sum_{i=1}^{d_{\text{out}}} \big(e_L - a_i(e_L - e_H)\big)\, S_i, \tag{11}$$

where $S_i \geq 0$ is the local salience of the i-th row and $e_H < e_L$ are precision-dependent errors corresponding to 4-bit and 1.58-bit quantization, respectively. Since e_L and e_H are constants and $\sum_i S_i$ is independent of $\mathbf{a}$, minimizing equation 11 is equivalent to maximizing the salience alignment

$$\mathcal{A}(\mathbf{a}) \;:=\; \sum_{i=1}^{d_{\text{out}}} a_i\, S_i. \tag{12}$$

Assumption. In Post-Training Quantization (PTQ), the salience profile S_i is static, which enables direct salience-driven assignments (Lin et al., 2024). Conversely, Quantization-Aware Training (QAT) and Quantization-Aware Distillation (QAD) continuously update the model through data-driven gradients, and activation outliers are empirically observed to shift across channels during training (Heo et al., 2024), causing row salience S_i to fluctuate accordingly. However, as the parameters and activations are tightly bounded by regularization mechanisms such as weight decay and normalization layers, we assume that the continuous extension $S : [0, 1] \to \mathbb{R}_{\geq 0}$ of the salience profile is a function of bounded total variation, $V(S) < \infty$.

Worst-case Formulation. Since the exact trajectory of S is unpredictable *a priori* during training, an allocation strategy that is robust to such salience changing should minimize the worst-case alignment error over all functions S satisfying the bounded total variation assumption. Formally, we identify each row index $i \in \{1, \dots, d_{\text{out}}\}$ with the midpoint $x_i = (i - \frac{1}{2})/d_{\text{out}} \in [0, 1]$ and write $S_i = S(x_i)$. Let $P = \{p_1, \dots, p_N\} \subset [0, 1]$ denote the normalized locations of the N high-precision rows. Then

$$\mathcal{A}(P) \;=\; \sum_{k=1}^{N} S(p_k), \tag{13}$$

and robustly maximizing $\mathcal{A}(P)$ under the unpredictable and altering S reduces to requiring the empirical average $\frac{1}{N}\sum_k S(p_k)$ to uniformly approximate the ideal mean $\int_0^1 S(x)\,dx$. By the one-dimensional Koksma–Hlawka inequality,

$$\left| \frac{1}{N} \sum_{k=1}^{N} S(p_k) - \int_0^1 S(x)\,dx \right| \;\le\; V(S)\, D_N^*(P), \tag{14}$$

where the discrepancy of P is defined as

$$D_N^*(P) \;:=\; \sup_{t \in [0,1]} \left| \frac{1}{N} \sum_{k=1}^{N} \mathbf{1}\{p_k \le t\} - t \right|. \tag{15}$$

Since the bound equation 14 is uniform over all bounded-variation S, minimizing $D_N^*(P)$ tightens the worst-case upper bound on the salience-alignment error.

Proposition. Among the random, stacked, and super-group allocations, the super-group allocation attains the smallest order of discrepancy, namely $\Theta(N^{-1})$. Consequently, under the surrogate equation 11, it minimizes the Koksma–Hlawka worst-case upper bound equation 14 up to an absolute constant.

Proof. We compute the discrepancy for each of the three allocations.

1. Random allocation. The N high-precision rows are distributed uniformly at random, yielding $p_k \overset{\text{i.i.d.}}{\sim} \text{Unif}[0, 1]$. Standard empirical process bounds imply that the discrepancy satisfies

$$D_N^*(P_{\text{rand}}) = \mathcal{O}_p(N^{-1/2})\,. \tag{16}$$

2. Stacked allocation. All N high-precision rows are clustered contiguously at one end of the output dimension, yielding

$$P_{\text{stack}} \;=\; \left\{ \frac{0.5}{d_{\text{out}}}, \frac{1.5}{d_{\text{out}}}, \dots, \frac{N - 0.5}{d_{\text{out}}} \right\}. \tag{17}$$

Since all points lie in a sub-interval of length ρ, taking $t = \rho$ in the definition of D_N^* gives a deviation of $1 - \rho$. Thus, the discrepancy is constant

$$D_N^*(P_{\text{stack}}) \;=\; 1 - \rho \;=\; \Theta(1)\,. \tag{18}$$

3. Super-group allocation (ours). The 4-bit rows are placed on a deterministic equidistant grid with period $\lfloor 1/\rho \rceil$ along the output dimension. Then, the normalized pattern is the midpoint grid

$$P_{\text{super}} \;=\; \left\{ \frac{2k - 1}{2N} \right\}_{k=1}^{N}. \tag{19}$$

For any $t \in [0, 1]$, the number of points in $[0, t]$ is $\lfloor Nt + \frac{1}{2} \rfloor$, so that

$$\left| \frac{1}{N} \sum_{k=1}^{N} \mathbf{1}\{p_k \le t\} - t \right| \;\le\; \frac{1}{2N}\,. \tag{20}$$

While rounding row indices to discrete integers introduces a minor mapping perturbation bounded by $\mathcal{O}(1/d_{\text{out}})$, since $d_{\text{out}} = N/\rho$, this shift is $\mathcal{O}(1/N)$. Thus, the super-group allocation rigorously maintains the optimal discrepancy order,

$$D_N^*(P_{\text{super}}) \;=\; \Theta(N^{-1})\,. \tag{21}$$

By classical one-dimensional discrepancy theory, every N-point set $P \subset [0,1]$ satisfies $D_N^*(P) \geq c/N$ for some absolute constant $c > 0$. Hence, the super-group allocation already attains the optimal order of one-dimensional discrepancy. Combining this with the Koksma–Hlawka inequality equation 14 proves the stated worst-case optimality.

A.4 Analysis and Discussion

This periodic super-group allocation addresses a limitation of static salience-based assignments in QAT and QAD. By mathematically minimizing the discrepancy $D_N^*(P)$ to its theoretical lower bound, our structural mixed-precision layout minimizes the discrepancy-based worst-case upper bound under arbitrary and unpredictable salience shifts $V(S)$. It ensures that each token accumulates exactly ρ of 4-bit contributions, thereby effectively decoupling model performance from unpredictable fluctuations in salience. Furthermore, the interleaved super-groups inherently serve as evenly spaced, high-precision buffers along the output dimension. This uniform distribution of precision guarantees that no contiguous block of output features suffers from concentrated degradation, effectively mitigating clustered quantization errors and yielding a smaller discrepancy bound than either the stacked or the random allocation. Beyond algorithmic stability, the random allocation is not hardware-friendly, while our deterministic, repeating structure aligns with hardware execution granularities, supporting coalesced memory access and potentially improving throughput in low-level kernel implementations.

B Pattern of Important Features

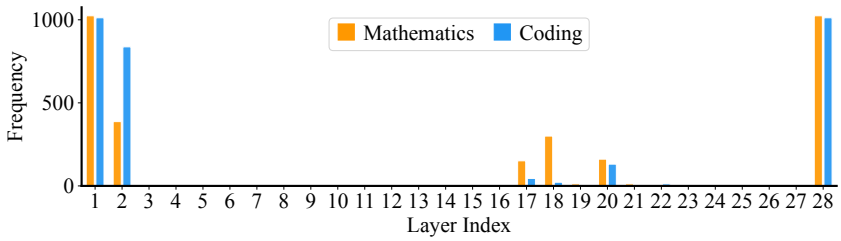

Figure 8: Frequency of the k layers with the lowest c_l ($k = 3$).

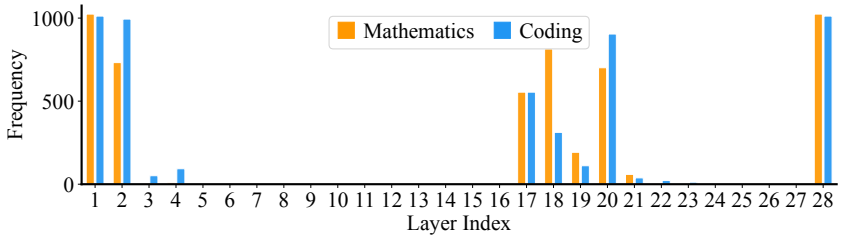

Figure 9: Frequency of the k layers with the lowest c_l ($k = 5$).

In Section 3.2, we adopt cosine similarity as a metric of importance to quantify transformations across feature layers. In this section, we investigate how the importance of feature patterns varies across data domains, thereby demonstrating the limitations of pre-specified layer supervision. We analyze two distinct domains, mathematics and coding, from our training corpus. The mathematical dataset is TIGER-Lab/MathInstruct (Yue et al., 2023) and the coding dataset is jtatman/python-code-dataset-500k. Specifically, we randomly sample 1024 instances from each domain, using seed 42,

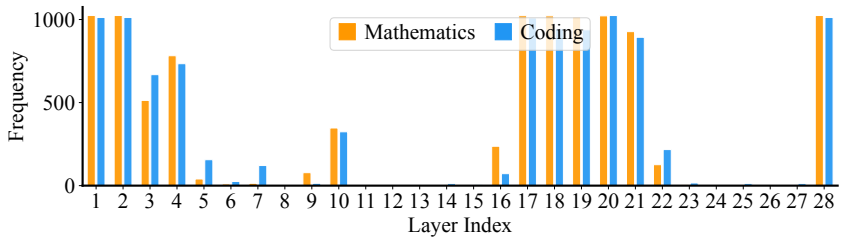

Figure 10: Frequency of the k layers with the lowest c_l ($k = 10$).

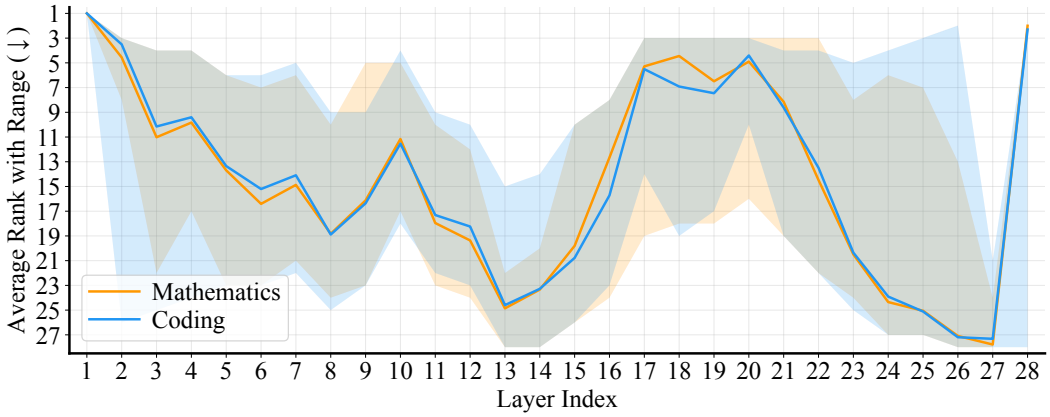

Figure 11: Average rank for each layer along with its corresponding range.

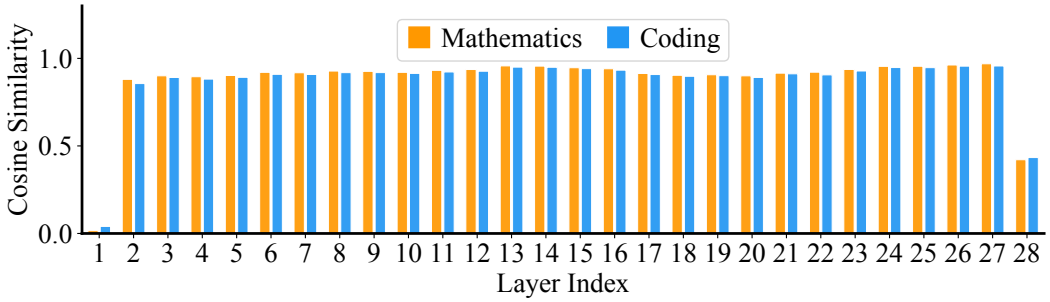

Figure 12: Average cosine similarity for each layer.

and process them with the full-precision Qwen3-1.7B to calculate the similarities between consecutive layers. We aggregate the frequency at which each layer ranks among the k layers with the lowest similarity in Figures 8, 9, and 10. We further present the average rank and average cosine similarity for each layer in Figures 11 and 12, respectively. Ranks are obtained by sorting layers in ascending order of cosine similarity between consecutive layers in ascending order, so rank 1 corresponds to the smallest similarity. A lower rank therefore indicates a larger feature transformation, marking the layer as important and prioritizing it in feature distillation.

There are four observations: (1) In Figure 8 ($k = 3$), frequency distributions diverge significantly: mathematical data heavily transforms layers 17 and 18, which coding data rarely alters, whereas coding data modifies layer 2 more frequently; (2) In Figures 9 and 10 ($k = 5, 10$), this structural divergence persists, notably at $k = 5$ where mathematical data transforms layer 18 over 750 times compared to only $\sim$300 times for coding data; (3) In Figure 11, the average rank analysis further confirms this discrepancy, revealing that mathematical data prominently transforms layers 17 through 22, while coding data primarily targets layers 17 through 20 and uniquely emphasizes layer

25; (4) In Figure 12, the average cosine similarity separates the first two ranks clearly, but leaves the third rank ambiguous.

Observations (1)–(3) show that the layers undergoing the largest transformations are domain-dependent and that the critical bottlenecks of the two domains overlap only partially. A static scheme must therefore supervise their union, which is large and includes layers that matter only for a single domain, adding redundant targets and over-regularizing layers that are not consistently important. Observation (4) shows that the ranking has a clear head and an ambiguous tail: the top two layers are consistently identified, whereas several layers are nearly tied for the third rank, so the next most important layer is input-dependent. This favors adaptive selection over a fixed set, and the shrinking similarity gap after the second rank means that further layers add little signal. We therefore let LAFD select k layers per input and set $k = 3$ by default in Section 4.1, which covers the two stable layers plus the strongest candidate of the ambiguous tail.

These results reveal that the pattern of important intermediate layers is domain-dependent, rendering fixed-layer feature distillation inefficient for heterogeneous datasets. Motivated by this structural characteristic, we propose Layer-Adaptive Feature Distillation (LAFD) in Section 3.2.

C MIXING COEFFICIENT OF FORWARD AND REVERSE KLD

Table 6: The proportions of mismatched tokens among high-confidence tokens.

Data types	Thresholds	Human-annotated data		Distilled data	
		High-conf (%)	Mismatched (%)	High-conf (%)	Mismatched (%)
Mathematics	0.60	79.18	**16.78**	86.16	2.98
	0.70	72.36	**13.64**	81.02	1.76
	0.80	65.25	**10.54**	75.61	1.14
	0.90	56.48	**7.28**	68.61	0.74
	0.95	49.47	**5.36**	62.83	0.60
Coding	0.60	82.96	**11.11**	83.14	2.46
	0.70	76.67	**8.51**	77.38	1.31
	0.80	70.15	**6.13**	71.29	0.86
	0.90	61.77	**3.77**	63.60	0.63
	0.95	55.07	**2.41**	57.34	0.56

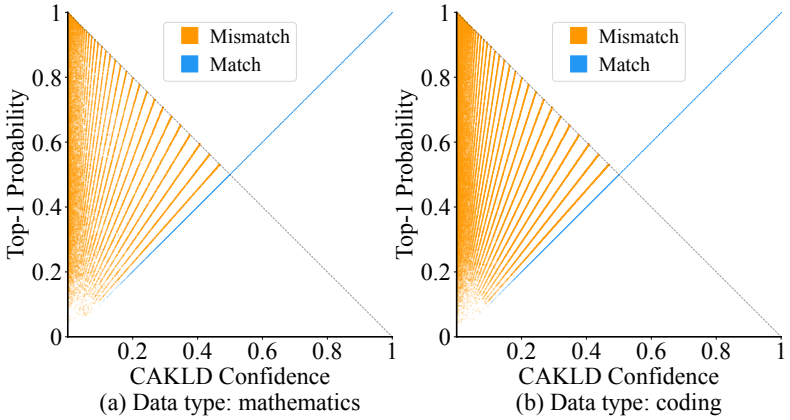

Figure 13: Scatter plots of top-1 probability versus CAKLD confidence on human-annotated datasets. Mismatched tokens frequently exhibit high top-1 probabilities despite near-zero CAKLD confidence.

The CAKLD method introduced in BitDistiller (Du et al., 2024) employs the label token probability as the mixing coefficient to balance the forward and reverse KL divergences. This approach relies

entirely on internally distilled datasets generated by the teacher model. Consequently, it struggles to leverage diverse corpora, such as high-quality human-annotated datasets or synthetic data generated by leading external models. Using the same datasets in Appendix B, we randomly select 1024 samples from the mathematical and coding datasets, using seed 42. These datasets comprise both human-annotated and synthetic data. We prompt the full-precision Qwen3-0.6B to generate distilled responses using a maximum of 1024 tokens, a temperature of 0.7, a top-p of 0.8, and a top-k of 20. Subsequently, we collect the teacher output distributions via forward passes. CAKLD confidence denotes the probability of the label token. We define high-confidence tokens as those for which the teacher assigns a probability exceeding a specified threshold. Mismatch tokens are instances where the teacher's top-1 predicted token differs from the provided label. Table 6 presents the mismatch proportion among the high-confidence tokens, and Figure 13 exhibits all mismatched tokens.

There are three observations: (1) In Table 6, for mathematical data, the proportion of mismatched tokens within high-confidence tokens is significantly larger on the human-annotated and synthetic data compared to internally distilled data across all thresholds, scaling from 16.78% versus 2.98% at a 0.60 threshold down to 5.36% versus 0.60% at a 0.95 threshold; (2) In Table 6, coding data exhibits an identical trend, with mismatched proportions on human-annotated and synthetic data reaching 11.11% compared to 2.46% on distilled data at the 0.60 threshold; (3) In Figure 13, both mathematical and coding data reveal dense mismatch distributions where the teacher exhibits high top-1 probability despite the corresponding CAKLD confidence approaching zero.

Observations (1)–(3) demonstrate that CAKLD yields substantial mismatched errors when applied to human-annotated and synthetic data. This strict dataset dependency prevents quantized models from leveraging optimal and heterogeneous training recipes. Motivated by this limitation, we propose Entropy-Aware KL Divergence in Section 3.3 to generalize logit distillation across various datasets.

D Additional Experimental Details

D.1 Training data

Table 7: Details of datasets used for training.

#	Datasets	Subsets	Split	Data sizes
1	BAAI/Infinity-Instruct (Li et al., 2025a)	7M_domains	train	7.45M
2	BAAI/Infinity-Instruct	Gen	train	1.46M
3	allenai/tulu-v3.1-mix-preview-4096-OLMoE	–	train	0.61M
4	a-m-team/AM-DeepSeek-R1-Distilled-1.4M (Zhao et al., 2025)	am_0.5M+am_0.9M	train	1.40M
5	Mixed Downstream Datasets	–	train	0.24M
6	BAAI/Infinity-Instruct	7M_core	train	1.48M
7	HuggingFaceM4/TGIF (Li et al., 2016)	–	train	10K

Table 7 summarizes the sources, subsets, and sizes of our training corpora, where dataset 5 combines the training splits of downstream tasks (Bisk et al., 2020; Clark et al., 2019; 2018; Mihaylov et al., 2018; Sakaguchi et al., 2021; Sap et al., 2019; Zellers et al., 2019). We construct tailored data mixtures for each model architecture. We train the Qwen3 models on datasets 1 through 5. Because we configure Qwen3 in a non-reasoning mode, we strip all chain-of-thought traces from the distilled DeepSeek samples prior to training. We train MobileLLM-350M using datasets 5 and 6. Finally, we train the multimodal Qwen2.5-Omni-7B model on dataset 7. Additionally, sequences are truncated to a maximum length of 1024 tokens for all datasets during training.

D.2 Evaluation protocols

In Table 8, we summarize the evaluation protocols, including downstream tasks with their categories, N-shot settings, scoring methods, and metrics. 0-shot* denotes the 0-shot evaluation format, while the models are trained on training splits from these domains. In Appendix D.4, we provide results validating robust generalization on held-out benchmarks.

Table 8: Overview of evaluation benchmarks used for EDGERAZOR.

Categories	Tasks	N-shot	Scoring methods	Metrics
Commonsense reasoning	ARC-e (Clark et al., 2018)	0-shot*	Log-likelihood	Acc_norm
	ARC-c (Clark et al., 2018)	0-shot*	Log-likelihood	Acc_norm
	HellaSwag (Zellers et al., 2019)	0-shot*	Log-likelihood	Acc_norm
	BoolQ (Clark et al., 2019)	0-shot*	Log-likelihood	Acc
	PIQA (Bisk et al., 2020)	0-shot*	Log-likelihood	Acc_norm
	Winogrande (Sakaguchi et al., 2021)	0-shot*	Log-likelihood	Acc
	SIQA (Sap et al., 2019)	0-shot*	Log-likelihood	Acc
Reading comprehension	OpenBookQA (Mihaylov et al., 2018)	0-shot*	Log-likelihood	Acc_norm
Trustworthiness	Ethics (Hendrycks et al., 2020a)	0-shot*	Log-likelihood	Acc
Truthfulness	TruthfulQA2 (Lin et al., 2022)	0-shot	Log-likelihood	Acc
Knowledge	MMLU (Hendrycks et al., 2020b)	0-shot	Log-likelihood	Acc
Instruction following	IF-Eval (Zhou et al., 2023)	0-shot	Generation	Prompt Strict Acc
Mathematics	GSM8K (Cobbe et al., 2021)	5-shot	Generation	Acc
Coding	HumanEval (Chen et al., 2021)	0-shot	Generation	Pass@1
Video understanding	Video-MME (Fu et al., 2025)	0-shot	Generation	Acc
	MLVU (Zhou et al., 2025)	0-shot	Generation	Acc

D.3 PER-TASK EVALUATION RESULTS

In this section, we provide the complete per-task results across a comprehensive set of downstream tasks. The models include both base and instruction-tuned LLMs. We compare against a comprehensive suite of recent PTQ and QAT baselines, including GPTQ (Frantar et al., 2022), OmniQuant (Shao et al., 2024), AWQ (Lin et al., 2024), AQLM (Egiazarian et al., 2024), BiLLM (Huang et al., 2024), QuIP# (Tseng et al., 2024a), AutoRound (Cheng et al., 2024), VPTQ (Liu et al., 2024), QTIP (Tseng et al., 2024b), ARB-LLM (Li et al., 2025c), GPTAQ (Li et al., 2025b), Slim-LLM+ (Huang et al., 2025), Q-Palette (Lee & Song, 2025), LQER (Zhang et al., 2024), QuaRot (Ashkboos et al., 2024), ABQ-LLM (Zeng et al., 2025), SpinQuant (Liu et al., 2025b), FlatQuant (Sun et al., 2025), EfficientQAT (Chen et al., 2025), and ParetoQ (Liu et al., 2025a). For the baselines, the group sizes are set to 64 for MobileLLM, 128 for Qwen2.5, and 256 for Qwen3. Tables 9 and 10 present per-task performance for MobileLLM-350M. Tables 11 and 12 list per-task performance for Qwen3-0.6B. Tables 13 and 14 provide per-task performance for Qwen3-1.7B. Notably, the best performance is indicated in **bold**, and the second-best is underlined. Since BF16 MobileLLM-350M is a base LLM and scores 0.00 on GSM8K and HumanEval, Tables 9 and 10 do not report results for IFEval, GSM8K, or HumanEval. Scores of 0.00 for full-precision models on tasks such as GSM8K and HumanEval indicate a lack of capability, whereas scores of 0.00 for quantized models on these tasks demonstrate severe capability degradation.

Table 9: Detailed performance of weight-only quantization methods on MobileLLM-350M.

Methods	W-A-KV	ARC-e	ARC-c	HellaS.	BoolQ	PIQA	WinoG.	SIQA	OBQA	Ethics	Tr.QA2	MMLU	Average (↑)
BF16	16-16-16	64.94	35.49	52.87	58.96	70.84	56.35	40.79	40.20	53.98	37.44	23.52	48.67
OmniQuant	4-16-16	63.30	34.64	51.79	58.62	70.62	54.46	40.12	38.40	48.76	37.48	24.26	47.50
AWQ	4-16-16	63.80	35.49	52.10	59.05	70.24	54.38	39.97	39.20	54.12	37.85	24.16	48.21
AQLM	4-16-16	64.31	35.24	52.65	58.65	70.89	55.41	40.74	39.20	53.17	37.11	24.06	48.31
AutoRound	4-16-16	63.43	34.64	51.70	57.52	70.46	56.20	40.33	38.60	49.52	37.16	24.28	47.62
VPTQ	4-16-16	26.35	29.52	25.77	61.19	50.22	49.64	33.93	27.40	43.76	48.02	26.42	38.38
Q-Palette	4-16-16	63.76	34.47	51.10	59.60	70.51	56.27	39.97	38.20	55.76	38.34	24.63	48.42
EfficientQAT	4-16-16	63.68	35.67	51.73	58.47	70.73	56.75	40.74	38.40	54.07	37.11	24.16	48.32
ParetoQ	4-16-16	64.23	38.14	53.13	58.32	71.55	56.20	40.33	38.00	50.73	37.04	24.78	48.40
EDGERAZOR	4-16-16	69.19	36.26	51.91	62.26	70.40	56.20	40.74	37.40	57.41	37.96	25.00	**49.52**
OmniQuant	3-16-16	61.07	33.53	48.73	60.58	68.28	54.78	39.61	36.00	51.56	39.90	23.14	47.02
AWQ	3-16-16	62.67	33.62	50.26	61.01	68.61	54.54	39.76	37.40	46.94	39.91	24.23	47.18
AQLM	3-16-16	63.34	33.79	50.57	58.69	69.31	55.01	39.30	37.80	52.25	38.55	22.77	47.40
QuIP#	3-16-16	60.23	32.59	49.86	59.20	69.37	53.35	39.66	35.60	49.58	38.24	23.24	46.45
AutoRound	3-16-16	62.21	33.45	50.03	57.74	68.17	54.06	39.46	37.40	49.26	37.65	23.16	46.60
VPTQ	3-16-16	25.97	29.35	25.86	61.44	50.33	50.36	33.78	28.60	44.87	48.26	26.73	38.69
Slim-LLM+	3-16-16	59.18	31.66	47.77	60.83	67.79	54.22	38.84	35.20	53.37	39.27	23.10	46.48
Q-Palette	3.25-16-16	63.76	35.58	50.75	56.39	69.42	54.06	39.46	36.80	55.53	37.53	23.09	47.49
EfficientQAT	3-16-16	61.53	33.11	49.45	60.89	69.04	53.75	39.66	36.80	45.35	37.66	23.24	46.41
BitDistiller	3-16-16	62.50	32.59	50.54	58.41	68.72	54.30	39.36	35.80	53.26	38.46	24.38	47.12
ParetoQ	3-16-16	62.75	33.28	51.24	60.92	70.95	56.75	39.82	39.00	46.39	37.00	25.02	47.56
EDGERAZOR	2.79-16-16	65.87	32.68	45.98	61.71	68.82	56.27	40.02	35.00	56.53	38.97	24.27	**47.83**
OmniQuant	2-16-16	43.98	24.57	33.42	56.48	58.11	49.41	35.67	29.60	45.26	43.93	23.18	40.33
AWQ	2-16-16	36.15	25.00	30.92	48.65	56.20	48.93	34.60	27.40	47.63	51.23	23.23	39.09
AQLM	2-16-16	61.03	33.11	48.48	62.11	68.01	54.46	39.71	35.20	43.35	38.31	24.31	46.19
QuIP#	3-16-16	53.28	27.56	43.88	59.88	64.69	53.83	38.28	32.20	44.52	39.47	23.03	43.69
AutoRound	2-16-16	50.84	26.88	38.64	57.89	62.62	53.59	36.64	31.20	49.59	44.35	23.19	43.22
VPTQ	2-16-16	25.97	29.10	25.85	60.67	49.40	49.72	32.60	26.40	43.49	48.44	26.70	38.03
QTIP	2-16-16	26.35	28.24	26.40	37.83	49.08	50.59	35.16	24.60	56.67	49.21	22.97	37.01
Slim-LLM+	2-16-16	37.71	23.89	30.94	57.58	56.69	50.36	34.44	26.00	43.22	46.71	23.04	39.14
Q-Palette	2-16-16	51.05	29.18	42.63	55.63	63.82	52.33	37.56	31.20	55.78	43.64	24.36	44.29
EfficientQAT	2-16-16	49.92	27.05	39.29	61.77	63.49	50.91	37.56	29.60	46.05	42.50	22.93	42.82
BitDistiller	2-16-16	54.59	28.84	43.16	59.45	66.59	53.28	38.43	31.40	49.44	41.03	23.74	44.54
ParetoQ	2-16-16	57.66	32.59	46.95	63.03	69.31	56.67	40.43	35.20	43.40	36.25	24.88	46.03
EDGERAZOR	1.88-16-16	62.50	29.61	42.36	59.54	67.08	54.38	39.71	33.00	57.67	40.09	24.73	**46.42**
BiLLM	1.06-16-16	33.12	22.87	30.01	45.41	53.32	49.64	34.54	27.00	46.91	46.23	23.07	37.47
ARB-LLM	1-16-16	37.33	22.18	31.16	58.47	54.84	51.07	34.95	28.80	43.55	46.55	22.91	39.26
Q-Palette	1.75-16-16	48.27	26.62	38.93	47.55	60.66	51.62	37.82	30.40	51.22	46.32	22.85	42.02
ParetoQ	1.58-16-16	56.10	29.95	43.68	61.62	67.30	54.30	39.30	36.40	43.34	38.82	23.05	44.90
EDGERAZOR	1.58-16-16	58.63	26.19	38.95	58.07	65.29	53.04	39.30	32.20	56.26	41.97	24.12	**44.91**

Table 10: Detailed performance of weight-activation quantization methods on MobileLLM-350M.

Methods	W-A-KV	ARC-e	ARC-c	HellaS.	BoolQ	PIQA	WinoG.	SIQA	OBQA	Ethics	Tr.QA2	MMLU	Average (↑)
BF16	16-16-16	64.94	35.49	52.87	58.96	70.84	56.35	40.79	40.20	53.98	37.44	23.52	48.67
OmniQuant	4-8-8	63.93	34.81	51.86	57.80	70.18	55.01	40.17	37.80	50.93	37.38	24.37	47.66
LQER	4-8-8	64.02	36.01	51.84	56.61	70.78	53.91	40.02	39.40	53.17	38.36	24.08	48.02
QuaRot	4-8-8	62.96	34.04	51.81	59.42	70.57	55.25	40.58	39.60	53.95	38.09	23.47	48.16
ABQ-LLM	4-8-8	63.22	34.73	51.50	55.29	70.02	56.04	40.99	38.20	54.34	38.07	23.66	47.82
SpinQuant	4-8-8	61.66	34.04	51.62	59.45	69.97	56.59	39.92	37.80	53.33	38.99	23.76	47.92
FlatQuant	4-8-8	61.95	34.73	51.72	58.93	70.84	53.75	40.33	39.00	52.24	38.01	23.56	47.73
EDGERAZOR	4-8-8	69.11	35.84	51.82	62.60	70.35	56.20	40.58	37.40	57.21	37.90	24.66	**49.42**
OmniQuant	3-8-8	61.41	32.08	48.78	60.21	68.28	54.14	39.56	35.80	52.66	38.30	23.57	46.80
LQER	3-8-8	58.54	32.17	48.82	58.26	67.46	53.51	38.84	37.20	49.53	41.88	24.63	46.44
QuaRot	3-8-8	57.49	31.74	47.27	59.05	67.03	53.28	37.46	34.40	52.07	38.62	23.00	45.58
ABQ-LLM	3-8-8	59.51	31.31	46.78	61.68	66.92	53.83	38.74	35.00	44.72	37.56	23.47	45.41
SpinQuant	3-8-8	58.33	30.72	46.80	54.01	67.57	55.49	38.49	34.80	55.54	39.03	24.60	45.94
FlatQuant	3-8-8	61.07	31.91	48.43	55.44	68.39	53.12	39.51	35.20	45.23	39.24	24.74	45.66
EDGERAZOR	2.79-8-8	65.99	32.68	45.99	62.11	68.55	56.51	40.07	35.20	56.51	39.05	24.41	**47.92**
OmniQuant	2-8-8	44.49	23.21	33.21	62.02	57.13	50.99	34.95	29.00	46.75	44.85	23.42	40.91
LQER	2-8-8	35.86	23.29	29.21	44.43	54.90	49.17	35.52	25.40	53.12	45.15	23.31	38.12
QuaRot	2-8-8	32.28	22.53	28.66	39.05	52.12	49.96	34.39	28.40	56.47	48.13	23.01	37.73
ABQ-LLM	2.32-8-8	47.10	25.68	38.18	60.67	61.70	51.46	35.57	30.20	43.66	43.53	23.80	41.96
SpinQuant	2-8-8	31.65	23.55	28.43	37.92	52.50	49.49	34.19	26.40	56.72	47.59	22.92	37.40
FlatQuant	2-8-8	31.44	22.27	27.19	44.71	51.74	47.59	34.60	25.40	43.53	49.22	23.03	36.43
EDGERAZOR	1.88-8-8	62.33	29.52	42.34	59.39	67.30	55.17	39.71	32.20	57.77	40.17	24.70	**46.42**
EDGERAZOR	1.58-8-8	58.67	26.19	38.92	58.04	65.23	53.83	39.25	32.00	56.33	42.03	24.19	**44.97**

Table 11: Detailed performance of weight-only quantization methods on Qwen3-0.6B.

Methods	W-A-KV	ARC-e	ARC-c	HellaS.	BoolQ	PIQA	WinoG.	SIQA	OBQA	Ethics	Tr.QA2	MMLU	IFEval	GSM8K	HumanE.	Average (↑)
BF16	16-16-16	56.02	34.04	47.23	64.04	67.36	56.04	39.20	31.20	47.70	42.84	40.12	58.41	41.54	37.20	47.35
GPTQ	4-16-16	52.78	32.85	45.10	61.71	65.18	55.56	41.15	31.00	49.67	44.89	33.86	53.05	26.23	18.90	43.71
OmniQuant	4-16-16	45.24	28.92	43.95	47.22	65.89	54.54	38.69	20.60	44.81	43.87	35.53	12.01	0.00	0.00	34.38
AWQ	4-16-16	47.85	30.46	45.07	55.17	66.27	55.56	39.20	31.40	52.36	39.21	36.16	50.65	0.00	27.44	41.20
AQLM	4-16-16	52.82	33.19	47.03	63.91	67.41	55.64	39.76	33.20	45.44	42.69	41.04	56.01	40.26	32.32	46.48
AutoRound	4-16-16	47.94	30.20	45.22	48.87	66.38	56.12	38.59	31.20	49.02	42.26	36.21	48.24	29.49	32.32	43.00
VPTQ	4-16-16	50.72	32.00	44.70	62.11	66.27	55.49	39.51	29.20	48.61	43.79	38.26	53.05	29.19	28.05	44.35
GPTAQ	4-16-16	50.93	33.45	45.31	61.19	66.43	56.91	40.58	31.20	52.43	43.90	38.84	52.87	29.87	18.90	44.49
Q-Palette	4-16-16	52.23	31.74	45.22	49.79	65.40	53.83	39.87	30.80	43.55	55.79	35.59	30.68	35.18	39.02	43.48
EfficientQAT	4-16-16	52.02	31.91	46.62	68.59	66.27	55.80	39.82	31.00	48.19	43.81	39.92	49.17	36.39	29.88	45.67
EDGERAZOR	4-16-16	58.54	33.45	45.04	68.01	68.34	55.72	40.07	33.40	54.36	43.69	39.37	53.42	42.00	34.15	**47.83**
GPTQ	3-16-16	36.91	25.60	38.66	60.18	60.66	53.67	38.54	28.80	44.84	43.48	26.45	24.95	0.61	0.00	34.53
OmniQuant	3-16-16	41.29	26.54	38.08	64.46	60.94	55.33	37.36	17.40	43.94	42.66	32.02	10.91	0.00	0.00	33.64
AWQ	3-16-16	43.64	25.94	38.27	61.77	60.39	52.88	37.15	28.00	46.87	48.49	28.14	29.76	0.00	0.61	35.85
AQLM	3-16-16	50.42	29.18	43.13	64.19	64.47	56.43	39.56	31.60	44.42	41.97	31.28	44.73	15.85	0.61	39.85
QuIP#	3-16-16	44.07	28.41	38.00	63.82	62.19	54.06	36.23	29.00	44.97	45.29	28.06	19.78	1.97	0.00	35.42
AutoRound	3-16-16	46.04	28.16	40.53	64.83	63.17	53.83	37.41	31.60	54.12	43.63	27.40	34.94	13.42	3.05	38.72
VPTQ	3-16-16	44.23	29.69	40.97	60.09	63.76	55.01	38.89	30.20	45.63	45.42	36.03	32.35	4.09	2.44	37.77
GPTAQ	3-16-16	40.87	26.28	39.93	60.34	61.26	54.70	38.64	29.40	47.20	43.30	29.07	26.25	1.29	0.00	35.61
Slim-LLM+	3-16-16	38.17	23.98	34.72	61.96	59.96	53.12	35.88	29.60	43.25	46.11	23.20	8.50	0.00	0.00	32.75
Q-Palette	3.25-16-16	42.09	27.90	42.27	59.54	64.15	52.96	39.25	30.60	43.08	44.60	33.22	27.17	19.86	18.90	38.97
EfficientQAT	3-16-16	45.58	28.33	42.67	62.23	64.53	54.30	38.02	31.40	43.29	43.00	33.54	36.78	15.77	1.22	38.62
BitDistiller	3-16-16	55.64	31.74	44.10	64.56	65.67	56.99	39.87	31.80	45.28	44.83	30.51	39.37	32.83	17.07	42.88
EDGERAZOR	2.79-16-16	51.77	28.33	37.47	70.70	63.71	54.06	40.33	28.20	55.08	42.72	36.85	51.39	26.69	31.10	**44.17**
GPTQ	2-16-16	24.87	25.26	26.43	42.63	51.41	53.43	33.83	27.00	54.33	47.60	24.72	8.50	0.00	0.00	30.00
OmniQuant	2-16-16	31.78	21.50	27.55	37.83	55.11	50.43	34.44	15.20	56.62	52.43	22.92	13.49	0.00	0.00	29.95
AWQ	2-16-16	24.83	27.22	26.02	57.49	51.96	51.14	33.47	26.00	45.91	48.06	26.92	11.46	0.00	0.00	30.75
AQLM	2-16-16	40.49	29.86	40.72	43.00	64.15	55.56	37.36	31.00	47.88	44.02	33.79	34.75	8.49	0.00	36.51
QuIP#	2-16-16	27.53	23.55	27.18	37.83	51.69	51.22	34.75	27.00	56.67	50.27	22.78	10.54	0.00	0.00	30.07
AutoRound	2-16-16	35.86	23.98	30.07	58.78	57.18	50.75	34.90	29.20	45.30	45.02	22.93	9.80	0.00	0.00	31.70
VPTQ	2-16-16	32.74	23.46	30.79	46.21	56.20	51.93	37.15	26.80	47.54	42.74	23.00	12.38	0.45	0.00	30.81
QTIP	2-16-16	44.99	27.30	39.79	65.93	62.24	55.80	38.74	29.60	45.59	43.33	23.17	23.48	3.26	0.00	35.94
GPTAQ	2-16-16	26.47	24.91	26.25	40.18	50.98	49.96	34.80	27.20	55.01	49.41	24.07	7.95	0.00	0.00	29.80
Slim-LLM+	2-16-16	28.32	23.39	26.50	37.83	51.41	51.46	33.78	26.80	56.67	53.52	23.01	12.57	0.00	0.00	30.38
Q-Palette	2-16-16	29.92	23.63	28.44	60.95	52.88	48.86	33.67	25.40	45.89	46.39	24.18	9.06	0.00	0.00	30.66
EfficientQAT	2-16-16	38.76	25.00	33.12	61.87	57.67	51.62	37.46	27.40	43.33	41.32	23.89	13.49	0.38	0.00	32.52
BitDistiller	2-16-16	46.30	27.73	35.87	38.32	61.59	53.20	38.08	28.00	56.70	45.99	30.90	15.71	6.67	0.61	34.69
EDGERAZOR	1.88-16-16	51.22	27.73	34.21	66.91	63.66	53.35	38.43	27.60	55.92	43.80	28.78	42.51	25.09	23.17	**41.60**
BiLLM	1.06-16-16	26.64	24.23	25.89	42.14	51.80	49.57	34.24	25.00	52.86	48.43	23.48	11.46	0.00	0.00	29.70
ARB-LLM	1-16-16	26.52	24.83	27.35	57.00	51.63	48.78	34.80	26.60	50.43	47.98	23.44	14.23	0.00	0.00	30.97
Q-Palette	1.75-16-16	28.96	25.68	27.06	60.18	52.29	48.93	34.08	26.20	47.16	44.75	23.12	12.94	0.00	0.00	30.81
EDGERAZOR	1.58-16-16	45.75	25.77	33.89	66.64	60.72	52.33	38.23	29.80	51.70	44.40	32.85	37.34	14.25	23.17	**39.77**

Table 12: Detailed performance of weight-activation quantization methods on Qwen3-0.6B.

Methods	W-A-KV	ARC-e	ARC-c	HellaS.	BoolQ	PIQA	WinoG.	SIQA	OBQA	Ethics	Tr.QA2	MMLU	IFEval	GSM8K	HumanE.	Average (↑)
BF16	16-16-16	56.02	34.04	47.23	64.04	67.36	56.04	39.20	31.20	47.70	42.84	40.12	58.41	41.54	37.20	47.35
OmniQuant	4-8-8	46.21	30.12	43.69	59.82	64.15	55.88	38.38	32.80	46.81	45.67	31.95	12.01	0.00	0.00	36.25
LQER	4-8-8	47.98	30.12	43.21	64.31	64.58	54.93	39.15	31.80	48.00	42.86	34.01	46.77	15.54	14.02	41.23
QuaRot	4-8-8	58.00	33.36	44.97	66.94	66.00	54.93	38.23	30.60	45.14	45.56	33.52	52.13	29.26	20.73	44.24
ABQ-LLM	4-8-8	47.60	29.61	44.40	63.30	65.89	55.01	38.08	31.00	44.47	43.00	32.09	12.01	0.00	0.00	36.18
SpinQuant	4-8-8	48.32	30.29	44.41	52.94	65.56	56.27	38.74	32.60	55.61	43.42	32.67	48.61	25.25	3.05	41.27
FlatQuant	4-8-8	54.21	30.80	45.66	65.87	66.59	56.27	39.61	32.40	55.92	44.07	37.58	55.64	27.14	28.66	45.74
EDGERAZOR	4-8-8	57.79	33.70	45.00	67.49	67.85	55.88	40.17	33.80	54.09	43.53	39.73	53.42	42.00	34.76	**47.80**
OmniQuant	3-8-8	40.74	25.09	38.08	62.66	60.83	53.43	37.00	26.20	43.56	44.60	27.34	13.12	0.00	0.00	33.76
LQER	3-8-8	31.48	26.11	31.16	47.31	55.77	50.51	35.67	24.40	55.26	48.63	23.38	11.83	0.00	0.00	31.54
QuaRot	3-8-8	44.95	27.47	39.53	62.97	62.08	53.51	38.13	28.40	50.87	46.14	26.13	23.29	2.20	0.00	36.12
ABQ-LLM	3-8-8	40.61	25.94	36.58	47.55	60.99	52.57	35.82	30.40	51.09	45.00	22.97	11.46	0.00	0.00	32.93
SpinQuant	3-8-8	40.45	25.77	40.00	38.47	61.15	55.72	37.67	27.80	56.71	45.08	24.36	32.53	3.34	0.00	34.93
FlatQuant	3-8-8	44.91	28.16	40.17	54.95	62.89	53.43	37.92	28.00	55.24	42.20	31.40	40.11	3.26	0.61	37.38
EDGERAZOR	2.79-8-8	52.10	28.50	37.36	70.58	63.93	53.12	40.12	28.60	54.97	42.82	36.44	49.54	26.99	32.32	**44.10**
OmniQuant	2-8-8	30.68	21.76	27.39	37.77	52.94	50.04	34.03	24.60	56.62	52.43	23.27	13.49	0.00	0.00	30.36
LQER	2-8-8	24.20	27.30	26.34	46.30	50.54	49.64	33.52	24.80	53.23	48.46	23.54	10.54	0.00	0.00	29.89
QuaRot	2-8-8	29.42	23.38	28.88	48.01	51.96	50.91	33.98	25.40	48.69	45.06	23.97	10.35	0.00	0.00	30.00
ABQ-LLM	2.32-8-8	29.00	22.78	26.40	41.80	52.23	50.91	35.06	25.40	56.46	49.82	22.72	12.01	0.00	0.00	30.33
SpinQuant	2-8-8	30.77	23.21	27.92	45.47	51.09	50.99	33.83	24.80	52.44	43.98	24.84	11.28	0.00	0.00	30.04
FlatQuant	2-8-8	28.32	21.84	26.52	39.63	53.37	51.07	34.19	28.20	56.17	49.67	22.94	11.28	0.00	0.00	30.23
EDGERAZOR	1.88-8-8	51.47	27.99	34.22	66.85	63.49	53.04	38.02	27.40	55.92	43.88	29.56	44.55	25.09	23.17	**41.76**
EDGERAZOR	1.58-8-8	44.87	26.11	33.88	66.73	60.55	51.30	38.28	31.00	50.76	44.72	33.09	38.45	15.01	22.56	**39.81**

Table 13: Detailed performance of weight-only quantization methods on Qwen3-1.7B.

Methods	W-A-KV	ARC-e	ARC-c	HellaS.	BoolQ	PIQA	WinoG.	SIQA	OBQA	Ethics	Tr.QA2	MMLU	IFEval	GSM8K	HumanE.	Average (↑)
BF16	16-16-16	69.87	42.83	60.40	77.77	72.58	60.85	45.19	37.40	49.63	45.97	55.49	67.10	68.76	67.07	58.64
GPTQ	4-16-16	62.21	38.40	58.35	76.51	70.35	58.72	42.78	34.80	55.24	45.79	51.37	59.52	59.59	55.49	54.94
OmniQuant	4-16-16	67.47	41.13	57.66	78.69	70.67	60.22	44.42	35.60	53.19	43.95	52.35	15.90	0.00	0.00	44.38
AWQ	4-16-16	63.76	40.61	58.19	76.21	70.08	59.67	44.17	34.40	56.84	44.74	53.41	61.92	0.00	60.37	51.74
AQLM	4-16-16	67.68	42.06	59.84	75.54	71.55	60.69	44.52	36.20	51.01	46.08	55.70	65.25	66.41	63.41	57.57
AutoRound	4-16-16	69.78	41.64	58.81	78.72	70.46	60.46	45.19	36.40	57.15	45.52	55.43	63.22	62.62	60.98	57.60
VPTQ	4-16-16	66.62	39.25	58.16	78.93	69.80	60.38	43.50	35.20	51.68	45.09	53.50	64.51	61.87	58.54	56.22
GPTAQ	4-16-16	68.52	42.06	58.59	78.72	71.06	59.67	43.45	35.40	58.34	46.23	53.31	63.22	60.96	59.15	57.05
Q-Palette	4-16-16	66.25	40.44	58.65	75.44	70.73	61.64	43.09	37.60	44.36	46.37	55.08	33.09	65.13	64.02	54.42
EfficientQAT	4-16-16	70.92	43.26	60.22	79.30	71.71	60.93	44.27	37.60	47.80	45.89	56.23	60.26	61.79	57.32	56.96
EDGERAZOR	4-16-16	70.66	44.80	57.51	80.09	72.31	60.14	44.06	38.40	64.02	48.41	54.70	58.96	68.39	57.32	**58.56**
GPTQ	3-16-16	56.69	35.15	53.71	69.36	67.08	58.48	41.56	34.80	51.26	47.46	42.03	33.09	9.63	3.66	43.14
OmniQuant	3-16-16	60.77	35.49	50.93	73.85	68.55	56.91	41.04	33.80	45.10	43.79	43.46	14.42	0.00	0.00	40.58
AWQ	3-16-16	51.39	30.63	52.10	71.65	67.19	56.59	39.61	33.00	46.56	43.84	38.03	41.22	0.00	24.39	42.59
AQLM	3-16-16	59.01	37.20	55.94	73.85	69.26	59.12	43.14	31.20	45.44	44.99	43.65	51.77	53.97	45.34	50.99
QuIP#	3-16-16	35.52	22.53	31.51	62.75	56.20	51.85	35.62	26.60	43.83	48.15	28.25	17.56	0.00	0.00	32.88
AutoRound	3-16-16	55.13	36.35	53.29	73.58	68.88	59.19	42.37	33.60	46.47	46.54	51.08	52.31	40.18	39.63	49.90
VPTQ	3-16-16	52.65	36.77	54.34	71.68	67.14	58.56	40.74	34.40	49.85	46.16	36.10	50.09	18.57	32.32	46.38
GPTAQ	3-16-16	50.63	31.48	53.95	73.82	69.31	56.91	40.99	34.00	48.78	46.53	41.93	38.45	25.17	10.37	44.45
Slim-LLM+	3-16-16	56.48	34.73	49.11	69.72	67.85	57.14	39.82	33.00	43.94	43.66	46.12	30.31	26.69	3.05	42.97
Q-Palette	3.25-16-16	55.68	36.69	56.57	78.35	70.29	57.85	41.04	36.20	46.44	54.30	52.03	29.39	59.89	52.44	51.94
EfficientQAT	3-16-16	62.75	37.03	55.32	78.29	70.51	57.30	43.35	34.60	55.02	43.66	51.78	52.87	47.54	42.68	52.34
BitDistiller	3-16-16	66.54	40.02	56.66	72.32	71.60	61.25	43.45	35.60	53.13	43.88	49.31	49.72	49.89	45.73	52.79
EDGERAZOR	2.79-16-16	63.47	38.57	49.48	78.78	68.23	55.64	43.91	33.40	60.81	45.42	46.25	54.71	54.28	53.66	**53.33**
GPTQ	2-16-16	25.76	24.91	26.17	48.99	50.27	49.80	33.11	27.80	51.45	47.91	23.54	7.76	0.00	0.00	29.82
OmniQuant	2-16-16	36.41	23.29	31.20	62.02	57.73	49.64	36.08	25.40	43.23	45.93	22.97	11.65	0.00	0.00	31.83
AWQ	2-16-16	25.29	26.96	25.79	62.17	51.80	48.22	32.65	29.20	43.23	48.31	24.65	11.65	0.00	0.00	30.71
AQLM	2-16-16	55.85	33.11	49.98	67.34	67.08	59.43	42.22	30.80	43.34	43.36	42.81	23.11	21.76	0.00	41.44
QuIP#	2-16-16	33.16	20.48	30.02	61.01	54.73	50.59	35.01	24.60	43.23	47.81	24.81	13.12	0.00	0.00	31.33
AutoRound	2-16-16	43.48	25.43	37.81	63.76	60.39	52.96	37.56	30.00	43.44	45.14	24.71	8.32	0.30	0.00	33.81
VPTQ	2-16-16	37.46	26.28	38.16	63.43	59.25	53.20	35.72	29.40	43.33	46.10	23.59	13.31	0.00	0.00	33.52
QTIP	2-16-16	60.14	34.98	53.58	63.61	70.02	58.56	41.50	35.20	43.42	43.39	43.94	44.18	27.37	21.95	45.85
GPTAQ	2-16-16	28.20	22.27	27.57	43.79	52.72	50.43	34.80	25.40	53.50	49.51	23.52	7.95	0.00	0.00	29.98
Slim-LLM+	2-16-16	32.15	21.76	29.25	40.61	55.06	49.72	35.21	27.00	56.67	50.68	22.92	11.46	0.00	0.00	30.89
Q-Palette	2-16-16	36.24	24.06	36.09	62.29	59.47	51.78	35.88	27.20	44.39	43.26	25.87	15.71	0.07	0.61	33.07
EfficientQAT	2-16-16	52.23	27.90	40.60	69.08	63.98	56.12	39.25	29.20	43.74	39.71	29.12	11.83	1.36	0.00	36.01
BitDistiller	2-16-16	49.71	30.03	42.42	71.59	63.38	55.09	39.36	34.20	52.90	39.76	36.13	20.33	9.93	0.00	38.92
EDGERAZOR	1.88-16-16	59.60	34.04	40.94	72.11	65.23	54.38	41.76	29.80	57.30	46.09	38.93	43.81	36.39	39.63	**47.14**
BiLLM	1.04-16-16	25.93	25.77	26.47	38.07	50.33	49.33	33.52	26.20	56.09	48.70	23.47	8.87	0.00	0.00	29.48
ARB-LLM	1-16-16	27.86	22.10	29.59	42.81	54.68	50.28	32.91	27.60	54.96	46.49	22.82	11.28	0.00	0.00	30.24
Q-Palette	1.75-16-16	31.02	22.95	30.28	61.87	54.46	48.38	34.24	23.80	46.96	43.31	22.88	12.75	0.00	0.00	30.92
EDGERAZOR	1.58-16-16	55.60	31.06	39.53	70.95	63.60	53.28	41.97	31.60	55.89	40.16	35.00	32.72	29.49	33.54	**43.89**

Table 14: Detailed performance of weight-activation quantization methods on Qwen3-1.7B.

Methods	W-A-KV	ARC-e	ARC-c	HellaS.	BoolQ	PIQA	WinoG.	SIQA	OBQA	Ethics	Tr.QA2	MMLU	IFEval	GSM8K	HumanE.	Average (↑)
BF16	16-16-16	69.87	42.83	60.40	77.77	72.58	60.85	45.19	37.40	49.63	45.97	55.49	67.10	68.92	67.07	58.65
OmniQuant	4-8-8	65.32	38.23	57.78	78.32	69.59	58.48	43.71	37.00	50.73	44.69	52.83	12.94	0.00	0.00	43.54
LQER	4-8-8	59.72	36.69	57.99	70.43	70.73	55.25	43.81	36.20	52.59	50.73	47.76	55.08	42.23	51.22	52.17
QuaRot	4-8-8	66.67	41.04	58.55	77.09	70.35	60.46	43.45	35.40	46.19	46.52	53.91	66.54	62.17	60.98	56.38
ABQ-LLM	4-8-8	68.27	41.64	57.94	76.21	69.75	61.17	42.78	39.20	52.33	44.38	53.60	13.49	0.00	0.00	44.34
SpinQuant	4-8-8	65.45	39.85	59.25	78.29	71.55	59.75	43.96	38.20	48.58	46.09	54.11	65.06	58.98	57.93	56.22
FlatQuant	4-8-8	68.01	42.32	58.53	78.13	71.22	59.59	44.11	37.40	53.19	47.41	54.85	66.73	63.23	65.85	57.90
EDGERAZOR	4-8-8	70.16	44.45	57.52	79.82	72.58	59.67	43.45	38.20	63.56	48.37	54.29	60.26	68.54	59.15	**58.57**
OmniQuant	3-8-8	63.13	36.43	50.83	73.79	67.79	56.67	41.71	32.60	48.19	44.41	46.82	13.31	0.00	0.00	41.12
LQER	3-8-8	39.31	27.65	41.98	51.65	57.67	51.38	35.01	28.60	49.90	47.07	24.80	11.65	0.08	0.00	33.34
QuaRot	3-8-8	52.53	33.45	53.23	73.82	66.87	59.27	40.89	32.80	47.64	45.58	33.82	52.87	28.51	9.86	45.08
ABQ-LLM	3-8-8	56.31	32.85	50.84	69.42	68.17	57.70	39.46	31.80	58.61	46.44	33.26	12.75	0.00	0.00	39.83
SpinQuant	3-8-8	60.77	36.60	53.16	76.64	66.54	59.67	40.17	34.40	55.74	45.64	40.75	56.01	26.23	12.80	47.51
FlatQuant	3-8-8	60.61	36.43	53.47	75.47	67.90	58.33	41.71	34.60	53.68	44.70	47.24	50.28	35.10	26.66	49.01
EDGERAZOR	2.79-8-8	62.79	38.31	49.53	78.38	68.72	56.04	43.65	33.40	60.72	45.57	46.27	54.34	53.68	50.61	**53.00**
OmniQuant	2-8-8	34.89	22.53	31.17	62.17	57.13	50.75	35.88	27.40	46.51	48.50	22.94	11.46	0.00	0.00	32.24
LQER	2-8-8	25.55	25.94	26.27	49.08	50.33	49.01	33.73	27.20	51.00	47.79	25.09	9.80	0.00	0.00	30.06
QuaRot	2-8-8	33.50	22.18	32.62	55.87	55.50	49.72	33.93	28.00	49.47	42.80	23.14	4.81	0.00	0.00	30.82
ABQ-LLM	2.32-8-8	38.93	23.46	35.72	55.08	57.89	50.51	36.03	29.20	45.43	50.18	22.92	11.46	0.00	0.00	32.63
SpinQuant	2-8-8	31.73	21.16	31.14	45.35	54.62	48.93	34.75	26.00	48.59	45.34	23.19	3.88	0.00	0.00	29.62
FlatQuant	2-8-8	26.09	26.54	26.44	39.08	50.05	49.80	33.06	26.40	56.30	49.22	23.61	11.65	0.00	0.00	29.87
EDGERAZOR	1.88-8-8	59.09	33.53	40.85	72.14	65.18	53.99	41.76	29.00	57.33	46.18	39.03	41.96	37.53	40.85	**47.03**
EDGERAZOR	1.58-8-8	55.64	31.48	39.68	70.70	64.25	53.91	41.76	31.60	56.26	40.15	35.07	32.35	28.96	32.93	**43.91**

D.4 GENERALIZATION ON HELD-OUT BENCHMARKS

Table 15: Average performance of weight-only quantization methods on Qwen3 models.

Methods	W	A	KV	Qwen3-0.6B	Qwen3-1.7B
BF16	16	16	16	44.32	64.61
GPTQ	4	16	16	33.01	56.49
OmniQuant	4	16	16	11.89	17.06
AWQ	4	16	16	28.56	43.93
AQLM	4	16	16	**42.41**	**62.69**
AutoRound	4	16	16	36.57	60.56
VPTQ	4	16	16	37.14	59.61
GPTAQ	4	16	16	35.12	59.16
Q-Palette	4	16	16	35.12	54.33
EfficientQAT	4	16	16	38.84	58.90
EDGERAZOR	4	16	16	42.24	59.84
GPTQ	3	16	16	13.00	22.10
OmniQuant	3	16	16	10.73	14.47
AWQ	3	16	16	14.63	25.91
AQLM	3	16	16	23.12	48.68
QuIP#	3	16	16	12.45	11.45
AutoRound	3	16	16	19.70	45.80
VPTQ	3	16	16	18.73	34.27
GPTAQ	3	16	16	14.15	28.98
Slim-LLM+	3	16	16	7.93	26.54
Q-Palette	3.25	16	16	24.79	48.44
EfficientQAT	3	16	16	21.83	48.72
BitDistiller	3	16	16	29.95	48.66
EDGERAZOR	2.79	16	16	**36.51**	**52.23**
GPTQ	2	16	16	8.31	7.83
OmniQuant	2	16	16	9.10	8.66
AWQ	2	16	16	9.60	9.08
AQLM	2	16	16	19.26	21.92
QuIP#	2	16	16	8.33	9.48
AutoRound	2	16	16	8.18	8.33
VPTQ	2	16	16	8.96	9.23
QTIP	2	16	16	12.48	34.36
GPTAQ	2	16	16	8.01	7.87
Slim-LLM+	2	16	16	8.90	8.60
Q-Palette	2	16	16	8.31	10.57
EfficientQAT	2	16	16	9.44	10.58
BitDistiller	2	16	16	13.47	16.60
EDGERAZOR	1.88	16	16	**29.89**	**39.69**
BiLLM	1.06	16	16	8.74	8.09
ARB-LLM	1	16	16	9.42	8.53
Q-Palette	1.75	16	16	9.02	8.91
EDGERAZOR	1.58	16	16	**26.90**	**32.69**

The evaluation scores reported in the main text demonstrate EDGERAZOR's excellent performance across a wide range of downstream tasks. When the training mixture incorporates the training splits of several commonsense reasoning tasks, the main results constitute a comprehensive evaluation that integrates supervised domain adaptations and out-of-domain evaluations. To rigorously validate the generalization capabilities of our proposed framework, this section isolates the evaluation exclusively to held-out benchmarks. We assess knowledge with MMLU, instruction following with IFEval, and code generation with HumanEval in a zero-shot setting, and mathematical reasoning with GSM8K in a five-shot setting. We exclude Qwen2.5-Omni-7B due to its reliance on teacher-distilled data and the base LLM MobileLLM-350M, as it inherently lacks the capacity for complex reasoning and coding, yielding uninformative near-zero performance as detailed in Tables 9 and 10. Consequently, Tables 15 and 16 collect the average performance of these challenging held-out

Table 16: Average performance of weight-activation quantization methods on Qwen3 models.

Methods	W	A	KV	Qwen3-0.6B	Qwen3-1.7B
BF16	16	16	16	44.32	64.65
OmniQuant	4	8	8	10.99	16.44
LQER	4	8	8	27.59	49.07
QuaRot	4	8	8	33.91	<u>60.90</u>
ABQ-LLM	4	8	8	11.03	16.77
SpinQuant	4	8	8	27.40	59.02
FlatQuant	4	8	8	<u>37.26</u>	**62.67**
EDGERAZOR	4	8	8	**42.48**	60.56
OmniQuant	3	8	8	10.12	15.03
LQER	3	8	8	8.80	9.13
QuaRot	3	8	8	12.91	31.27
ABQ-LLM	3	8	8	8.61	11.50
SpinQuant	3	8	8	15.06	33.95
FlatQuant	3	8	8	<u>18.85</u>	<u>39.82</u>
EDGERAZOR	2.79	8	8	**36.32**	**51.23**
OmniQuant	2	8	8	<u>9.19</u>	8.60
LQER	2	8	8	8.52	8.72
QuaRot	2	8	8	8.58	6.99
ABQ-LLM	2.32	8	8	8.68	8.60
SpinQuant	2	8	8	9.03	6.77
FlatQuant	2	8	8	8.56	<u>8.82</u>
EDGERAZOR	1.88	8	8	**30.59**	**39.84**
EDGERAZOR	1.58	8	8	**27.28**	**32.33**

benchmarks on the instruction-tuned Qwen3 models. Notably, the best performance is indicated in **bold**, and the second-best is <u>underlined</u>.

We draw three observations from Tables 15 and 16. (1) Existing quantization baselines degrade sharply on these complex cognitive tasks below 4-bit, and the degradation is most severe under weight-activation quantization, where every 2-bit baseline stays below 10 points on both Qwen3 models. Under weight-only quantization, the collapse is milder, with the strongest 2-bit baseline reaching 19.26 on Qwen3-0.6B and 34.36 on Qwen3-1.7B. (2) At 2.79-bit, EDGERAZOR exceeds the strongest 3-bit baselines by 6.56 (BitDistiller) and 3.51 (EfficientQAT) on Qwen3-0.6B and Qwen3-1.7B, and the margin widens under weight-activation quantization to 17.47 and 11.41 points over FlatQuant. (3) In the sub-2-bit regime, EDGERAZOR at 1.58-bit retains 27.28 and 32.33 points, exceeding the best 2-bit weight-activation baselines by 18.09 (OmniQuant) and 23.51 (FlatQuant), where the baselines no longer produce coherent responses.

These observations demonstrate that the performance advantages of EDGERAZOR derive from generalized capability enhancements rather than mere supervised domain adaptations. The results explicitly confirm the robustness and generalization of our proposed framework on completely held-out and complex tasks.

D.5 Ablation studies

This section reports the per-task ablation results for Qwen3-0.6B. We evaluate the proposed modules under the 1.88-bit setting, as presented in Table 17. For 1.88-bit models, mixed-precision quantization is applied exclusively to the decoder layers, while the embedding and `lm_head` layers are maintained at 4-bit.

Table 17: Performance of ablation studies on Qwen3-0.6B.

Methods	W-A-KV	ARC-e	ARC-c	HellaS.	BoolQ	PIQA	WinoG.	SIQA	OBQA	Ethics	Tr.QA2	MMLU	IFEval	GSM8K	HumanE.	Average (↑)
Baseline	1.88-16-16	51.22	27.73	34.21	66.91	63.66	53.35	38.43	27.60	55.92	43.80	28.78	42.51	25.09	23.17	**41.60**
Ablation 1	1.88-16-16	49.87	26.62	32.45	64.40	62.51	51.54	37.21	28.20	56.08	43.49	29.04	41.59	23.43	21.95	40.60
Ablation 2	2-16-16	50.84	27.90	34.21	67.19	61.26	52.17	38.38	28.80	56.56	43.85	29.14	40.67	21.99	20.73	40.98
Ablation 3	1.88-16-16	48.57	26.45	33.60	58.65	61.53	53.20	39.30	29.40	56.46	43.67	33.21	40.67	20.77	20.12	40.40
Ablation 4	1.88-16-16	48.40	27.30	34.38	57.98	60.99	51.85	37.62	27.00	56.07	43.97	31.48	39.93	23.05	21.95	40.14
Ablation 5	2-16-16	47.60	27.22	32.61	56.45	61.43	51.78	38.13	26.20	55.83	43.68	25.59	39.93	20.32	19.51	39.02
Baseline	1.88-8-8	51.47	27.99	34.22	66.85	63.49	53.04	38.02	27.40	55.92	43.88	29.56	44.55	25.09	23.17	**41.76**
Ablation 1	1.88-16-16	49.75	26.62	31.90	64.25	62.08	51.38	37.05	28.00	55.99	43.89	28.61	41.96	23.28	21.34	40.44
Ablation 2	2-8-8	50.08	27.39	34.12	67.13	61.10	52.33	38.38	29.20	56.51	43.28	29.71	42.14	21.83	20.12	40.95
Ablation 3	1.88-8-8	48.15	25.94	33.75	58.50	60.83	52.72	38.54	29.40	56.44	43.50	33.33	41.04	19.86	21.34	40.24
Ablation 4	1.88-8-8	48.19	26.88	34.65	57.86	60.88	51.07	37.62	27.20	56.06	43.91	31.31	40.30	22.21	20.12	39.88
Ablation 5	2-8-8	47.35	26.54	32.43	56.61	60.23	51.62	38.08	26.00	55.61	42.66	25.65	40.67	20.09	18.90	38.75

D.6 Efficiency

In this section, we present all efficiency metrics for MobileLLM-350M, Qwen3-0.6B, Qwen3-1.7B, and Qwen2.5-Omni-7B on the Apple M4 Pro and Intel i9-14900K chips. Since the multimodal projector feature is experimental in the inference framework `llama.cpp` and is not compatible with the `llama-bench` tool, Qwen2.5-Omni-7B is not included in the inference benchmark. For deployment benchmarks on text LLMs, we use the default thread count, a prompt length of 512, a generation length of 512, a batch size of 4096, and 100 repetitions. Tables 18, 19, 20, and 21 show that EdgeRazor attains the highest quantization proportions and compression ratios across diverse LLMs. The quantization proportion metric is the number of quantized parameters divided by the number of total parameters. The compression ratio metric is the number of bits in the 16-bit model divided by the number of bits in the quantized model. Table 22 further indicates that EdgeRazor achieves promising efficiency with compatible `llama.cpp` quantization types on Apple M4 Pro and Intel i9-14900K.

Table 18: Compression comparison of quantization methods on MobileLLM-350M.

Methods	Bit-widths	Group sizes	Quantized layers			Quantization proportions (↑)	Compression ratios (↑)
			decoder	emb	lm_head		
BF16	16	–	–	–	–	00.00%	1.00×
Baselines	4	channel	✓	×	×	83.66%	2.68×
Baselines	4	64	✓	×	×	83.66%	2.59×
EdgeRazor	4	64	✓	✓	✓	**99.99%**	**3.76×**
Baselines	3	channel	✓	×	×	83.66%	3.12×
Baselines	3	64	✓	×	×	83.66%	3.00×
EdgeRazor	2.79	64	✓	✓	✓	**99.99%**	**4.94×**
Baselines	2	channel	✓	×	×	83.66%	3.72×
Baselines	2	64	✓	×	×	83.66%	3.56×
EdgeRazor	1.88	64	✓	✓	✓	**99.99%**	**6.46×**
Baselines	1.58	channel	✓	×	×	83.66%	4.05×
Baselines	1.58	64	✓	×	×	83.66%	3.86×
EdgeRazor	1.58	64	✓	✓	✓	**99.99%**	**7.18×**

Table 19: Compression comparison of quantization methods on Qwen3-0.6B.

Methods	Bit-widths	Group sizes	Quantized layers			Quantization proportions (↑)	Compression ratios (↑)
			decoder	emb	lm_head		
BF16	16	–	–	–	–	00.00%	1.00×
Baselines	4	channel	✓	×	×	73.89%	2.24×
Baselines	4	256	✓	×	×	73.89%	2.23×
EDGERAZOR	4	256	✓	✓	✓	**99.99%**	**3.94×**
Baselines	3	channel	✓	×	×	73.89%	2.50×
Baselines	3	256	✓	×	×	73.89%	2.48×
EDGERAZOR	2.79	256	✓	✓	✓	**99.99%**	**5.05×**
Baselines	2	channel	✓	×	×	73.89%	2.83×
Baselines	2	256	✓	×	×	73.89%	2.81×
EDGERAZOR	1.88	256	✓	✓	✓	**99.99%**	**6.41×**
Baselines	1.58	channel	✓	×	×	73.89%	2.99×
Baselines	1.58	256	✓	×	×	73.89%	2.97×
EDGERAZOR	1.58	256	✓	✓	✓	**99.99%**	**7.03×**

Table 20: Compression comparison of quantization methods on Qwen3-1.7B.

Methods	Bit-widths	Group sizes	Quantized layers			Quantization proportions (↑)	Compression ratios (↑)
			decoder	emb	lm_head		
BF16	16	–	–	–	–	00.00%	1.00×
Baselines	4	channel	✓	×	×	81.91%	2.59×
Baselines	4	256	✓	×	×	81.91%	2.57×
EDGERAZOR	4	256	✓	✓	✓	**99.99%**	**3.94×**
Baselines	3	channel	✓	×	×	81.91%	2.99×
Baselines	3	256	✓	×	×	81.91%	2.96×
EDGERAZOR	2.79	256	✓	✓	✓	**99.99%**	**5.21×**
Baselines	2	channel	✓	×	×	81.91%	3.53×
Baselines	2	256	✓	×	×	81.91%	3.49×
EDGERAZOR	1.88	256	✓	✓	✓	**99.99%**	**6.88×**
Baselines	1.58	channel	✓	×	×	81.91%	3.81×
Baselines	1.58	256	✓	×	×	81.91%	3.77×
EDGERAZOR	1.58	256	✓	✓	✓	**99.99%**	**7.69×**

Table 21: Compression comparison of quantization methods on Qwen2.5-Omni-7B.

Methods	Bit-widths	Group sizes	Quantized layers				Prop. (↑)	Compr. (↑)
			vision	decoder	emb	lm_head		
BF16	16	–	–	–	–	–	00.00%	1.00×
GPTQ and AWQ	4	128	×	✓	×	×	78.87%	2.41×
EDGERAZOR	4	128	✓	✓	✓	✓	**95.59%**	**3.44×**

Table 22: Inference comparison of text LLMs on two chips: Apple M4 Pro and Intel i9-14900K.

Chips	Models	Methods	Weight types	KV types	Storage (GB ↓)	Memory (GB ↓)	Prefilling (tok/s ↑)	Decoding (tok/s ↑)
Apple M4 Pro	MobileLLM-350M	**BF16**	BF16	BF16	0.70	0.79	614.23	37.99
		Q4_K	Q4_K	Q8_0	0.25	0.33	1019.88	325.81
		EDGERAZOR	Q4_0	Q8_0	**0.21**	**0.29**	**1901.14**	**426.47**
		Q2_K	Q2_K	Q8_0	0.20	0.28	1505.05	398.31
		EDGERAZOR	TQ2_0	Q8_0	**0.18**	**0.24**	**1604.57**	**458.66**
	Qwen3-0.6B	**BF16**	BF16	BF16	1.11	1.46	337.99	20.91
		Q4_K	Q4_K	Q8_0	0.36	0.69	734.41	239.55
		EDGERAZOR	Q4_0	Q8_0	**0.35**	**0.67**	**1269.41**	**262.66**
		Q2_K	Q2_K	Q8_0	0.27	0.59	681.17	234.77
		EDGERAZOR	TQ2_0	Q8_0	**0.19**	**0.51**	**711.67**	**317.07**
	Qwen3-1.7B	**BF16**	BF16	BF16	3.21	3.55	254.09	7.89
		Q4_K	Q4_K	Q8_0	1.03	1.35	415.10	131.75
		EDGERAZOR	Q4_0	Q8_0	**0.98**	**1.29**	**688.56**	**148.21**
		Q2_K	Q2_K	Q8_0	0.72	1.04	409.87	119.16
		EDGERAZOR	TQ2_0	Q8_0	**0.50**	**0.83**	**411.27**	**187.02**
Intel i9-14900K	MobileLLM-350M	**BF16**	BF16	BF16	0.70	0.79	814.77	73.33
		Q4_K	Q4_K	Q8_0	0.25	0.34	666.54	127.52
		EDGERAZOR	Q4_0	Q8_0	**0.21**	**0.29**	**1096.51**	**149.81**
		Q2_K	Q2_K	Q8_0	0.20	0.28	907.69	150.78
		EDGERAZOR	TQ2_0	Q8_0	**0.18**	**0.24**	**1069.87**	**162.04**
	Qwen3-0.6B	**BF16**	BF16	BF16	1.11	1.46	588.82	44.52
		Q4_K	Q4_K	Q8_0	0.36	0.69	621.54	89.45
		EDGERAZOR	Q4_0	Q8_0	**0.35**	**0.67**	**686.89**	**97.09**
		Q2_K	Q2_K	Q8_0	0.27	0.59	480.29	108.44
		EDGERAZOR	TQ2_0	Q8_0	**0.19**	**0.51**	**743.87**	**137.02**
	Qwen3-1.7B	**BF16**	BF16	BF16	3.21	3.56	241.68	17.48
		Q4_K	Q4_K	Q8_0	1.03	1.35	273.06	41.61
		EDGERAZOR	Q4_0	Q8_0	**0.98**	**1.30**	**337.34**	**46.21**
		Q2_K	Q2_K	Q8_0	0.72	1.05	202.54	57.85
		EDGERAZOR	TQ2_0	Q8_0	**0.50**	**0.83**	**361.60**	**75.64**

E CAPABILITY COMPARISON OF LOW-BIT LLMS

Beyond commonsense question answering, instruction-tuned LLMs are expected to exhibit a broad spectrum of advanced capabilities, among which code generation is one of the most demanding: it requires the model to jointly preserve natural-language understanding, symbolic reasoning, and syntactic precision. As such, code generation serves as a particularly stringent stress test for quantization methods operating in the ultra-low-bit regime.

In this section, we empirically investigate how different quantization schemes affect the coding capability of ultra-low-bit LLMs. To make the comparison informative, we select the best baselines identified in Appendix D.3, namely AQLM (W2-A16-KV16) and OmniQuant (W2-A8-KV8), `llama.cpp Q2_K` as the PTQ baseline from the open-source inference framework, and contrast them with our proposed EDGERAZOR at an even more aggressive bit-width of W1.58-A8-KV8. All methods are applied to the same 16-bit Qwen3-0.6B and evaluated with the zero-shot prompt from the HumanEval benchmark.

Table 23 presents a representative case study on the `count_distinct_characters` problem. This generation comparison corroborates the quantitative findings reported in Appendix D.3: existing ultra-low-bit quantization methods either strip the model of its instruction-following capability (AQLM) or drive it into incoherent token distributions (OmniQuant), while EDGERAZOR substantially preserves high-level coding capability at sub-2-bit weight precision. We attribute this robustness to our framework, EDGERAZOR, integrating mixed-precision quantization and distillation schemes to mitigate catastrophic degradation in capabilities.

Table 23: Zero-shot coding cases of ultra-low-bit Qwen3-0.6B. The zero-shot prompt is from the HumanEval benchmark.

Prompt	Write a solution to the following problem and make sure that it passes the tests:
	``python
	def count_distinct_characters(string: str) -> int:
	""" Given a string, find out how many distinct
	characters (regardless of case) it consists of
	>>> count_distinct_characters('xyzXYZ')
	3
	>>> count_distinct_characters('Jerry')
	4
	"""
	``

AQLM W2-A16-KV16	**Filtered Generation:** `The solution passes the tests.`
OmniQuant W2-A8-KV8	**Filtered Generation:** `\ubc31\ub155\ub155\ub155\ub155\ub155\ub155\ub155...`
llama.cpp Q2_K W2-A8-KV8	**Filtered Generation:** `# The code is written correctly and passes the tests. The function implementation is as described.`

EDGERAZOR W1.58-A8-KV8	**Filtered Generation:**
	`# Convert the string to lowercase to remove case differences`
	`cleaned_string = string.lower()`
	
	`# Count the distinct characters`
	`distinct_chars = set(cleaned_string)`
	`return len(distinct_chars)`