BoA: Attention-aware Post-training Quantization without Backpropagation

Junhan Kim [*1] **Ho-young Kim** [*1] **Eulrang Cho** [1] **Chungman Lee** [1] **Joonyoung Kim** [1] **Yongkweon Jeon** [1]

Abstract

Post-training quantization (PTQ) is a promising solution for deploying large language models (LLMs) on resource-constrained devices. Early methods developed for small-scale networks, such as ResNet, rely on gradient-based optimization, which becomes impractical for hyper-scale LLMs with billions of parameters. While recently proposed backpropagation-free or transformation-based methods alleviate this issue, they ignore inter-layer interactions or use the naïve nearest-rounding-based quantized weight assignment to save the heavy computational cost of weight optimization. In this paper, we introduce a novel backpropagation-free PTQ algorithm that optimizes quantized weights by considering inter-layer dependencies. The key innovation is the development of attention-aware Hessian matrices that capture inter-layer interactions within the attention module. Extensive experiments demonstrate that our approach not only outperforms existing weight quantization methods but also shows good synergy with conventional methods to suppress activation outliers, leading to state-of-the-art weight-activation quantization performance. The code will be available at https://github.com/SamsungLabs/BoA.

1. Introduction

The explosive growth in complexity (parameters) of large language models (LLMs) (Touvron et al., 2023a; Zhang et al., 2022) has resulted in a proportional increase in computational costs, which has prompted an urgent need for efficient model processing and compression strategies. Quantization has emerged as a pivotal solution and an essential step for deploying AI models on resource-constrained devices that primarily support fixed-point arithmetic.

Two main categories of quantization approaches have been proposed to preserve the performance of original models: quantization-aware training (QAT) (Jin et al., 2021; Liu et al., 2023) and post-training quantization (PTQ) (Nagel et al., 2020; Kim et al., 2024). Although QAT can potentially outperform PTQ, its practicality diminishes when handling hyper-scale LLMs featuring billions of parameters. Consequently, recent quantization efforts have been directed toward PTQ.

Although classic PTQ methods have successfully quantized small models such as ResNet (Nagel et al., 2020; Li et al., 2021; Jeon et al., 2022), they rely on time-consuming gradient-based optimization, so their efficacy decreases when the complexity of LLMs increases. Accordingly, recent efforts have shifted to develop (a) backpropagation-free methods that optimize quantized weights based on Hessian (Frantar et al., 2023) or (b) transformation-based methods that convert a model to be robust to quantization by applying smoothing, rotation, and permutation (Shao et al., 2023; Ashkboos et al., 2024; Liu et al., 2024; Lin et al., 2024). However, their weight quantization performance is limited as they ignore inter-layer dependencies or use naïve nearest rounding for weight quantization.

In this paper, we propose a novel backpropagation-free PTQ algorithm that considers inter-layer dependencies in optimizing quantized weights. Our contributions are as follows:

- We propose a novel PTQ algorithm called BoA[1]. The key contribution is to exploit the attention reconstruction error, not the layer-wise reconstruction error, in approximating the Hessian to consider inter-layer dependencies within the attention module (**Section 3.2**). To our knowledge, BoA is the first method to optimize quantized weights by considering inter-layer dependencies without relying on gradient-based optimization.
- While the proposed Hessian facilitates the consideration of inter-layer dependencies, it requires additional memory and computations. To mitigate the overhead, we propose several techniques, including Hessian relaxation, efficient computation of inverse Hessians, and

[*]Equal contribution [1]Samsung Research, Seoul, Republic of Korea. Correspondence to: Yongkweon Jeon <dragwon.jeon@samsung.com>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

[1]Backpropagation-free optimization for Attention-aware PTQ

head-wise simultaneous quantization (**Section 3.3**).

- We evaluate BoA on publicly available LLMs. From extensive experiments, we show that BoA outperforms existing backpropagation-free weight quantization methods by a significant margin, particularly for low-bit precision (*e.g.*, INT2) (**Section 4.2**). Furthermore, when combined with existing methods to suppress outliers (Ashkboos et al., 2024; Liu et al., 2024), BoA achieves the state-of-the-art performance for both weight-only and weight-activation quantization (**Sections 4.2 and 4.3**).

2. Related Works

When calibration data are available, PTQ aims to minimize the increase in task loss incurred by quantization. Assuming the convergence of a network and the independence between layers, the problem of quantizing weights to minimize task loss degradation can be formulated as the following layer-wise reconstruction problem (Nagel et al., 2020):

$$\min_{\Delta \mathbf{W}^{(\ell)}} \left\| \left(\mathcal{Q}(\mathbf{W}^{(\ell)}) - \mathbf{W}^{(\ell)} \right) \mathbf{X}^{(\ell-1)} \right\|_F^2, \tag{1}$$

where $\mathbf{X}^{(\ell-1)}$ is the input to the ℓ-th layer parameterized by $\mathbf{W}^{(\ell)}$ and $\mathcal{Q}$ is a quantization function. For a uniform quantization, if the nearest quantization bin is assigned to each weight, $\mathcal{Q}$ is defined as

$$\mathcal{Q}(x) = s \left(\text{clamp} \left(\left\lfloor \frac{x}{s} \right\rceil + z, 0, 2^n - 1 \right) - z \right),$$

where s, z, n are the scale, zero-point, and bit-width, respectively, and $\lfloor \cdot \rceil$ represents the round-off.

Early works aimed to optimize the weight-rounding mechanism (Nagel et al., 2020; Hubara et al., 2021; Li et al., 2021; Jeon et al., 2023a). Instead of allocating the nearest integer, these studies attempted to assign integer weights that minimize the reconstruction error. One popular approach is AdaRound, which learns quantized weights satisfying (1) via backpropagation (Nagel et al., 2020). This algorithm has been extended to BRECQ where the block-wise reconstruction error has been used, instead of the layer-wise reconstruction error, to consider the inter-layer dependencies (Li et al., 2021). Although AdaRound and BRECQ have successfully quantized small-sized models, they rely on time-consuming gradient-based optimizations, which renders their application to LLMs with billions of parameters challenging. Consequently, recent efforts have shifted towards the development of cost-effective quantization methods for LLMs.

These efforts can be classified into two orthogonal classes: **(a)** backpropagation-free Hessian-based integer weight optimization methods (*e.g.*, GPTQ (Frantar et al., 2023)) and **(b)** transformation-based methods that convert a model to be more robust to quantization by applying smoothing (*e.g.*, SmoothQuant (Xiao et al., 2023), OmniQuant (Shao et al., 2023), and AffineQuant (Ma et al., 2024)), rotation (*e.g.*, QuaRot (Ashkboos et al., 2024) and SpinQuant (Liu et al., 2024)), or permutation (*e.g.*, DuQuant (Lin et al., 2024)).

The proposed BoA belongs to the class (a) in that it optimizes quantized weights using the Hessian without relying on backpropagation. Furthermore, similar to GPTQ, BoA can be combined with transformation-based methods such as QuaRot and SpinQuant (see Section 4). The primary difference over GPTQ lies in our optimization objective: while GPTQ assumes layer-wise independence and focuses on layer-wise reconstruction (which leads to performance degradation), our approach explicitly aims to preserve the attention output, enabling us to account for inter-layer dependencies within the attention module.

Other algorithms exploiting different strategies have also been proposed. For example, SpQR (Dettmers et al., 2023), SqueezeLLM (Kim et al., 2023), and OAC (Edalati et al., 2024) proposed a mixed-precision approach that assigns a large bit-width to quantization-sensitive weights or retains them in full-precision. Compared to the standard uniform quantization, these algorithms require additional processing and memory costs in the inference and need special hardware and dedicated kernels without which accelerating the inference may not be easy. Furthermore, unlike server-grade GPUs, on-device NPUs (*e.g.* Qualcomm Hexagon) lack support for the mixed precision format, and customizing kernels for desired functionalities is very challenging. Thus, we exclude these algorithms in our comparison and focus on the more universally supported uniform quantization format. We also exclude recent vector quantization approaches (*e.g.* QuIP# (Tseng et al., 2024) and AQLM (Egiazarian et al., 2024)) because they need additional memory (bits) for storing codebooks, which are required to perform the dequantization during the inference.

3. Method

3.1. Overview of Proposed BoA

The proposed BoA quantizes weights by repeating quantization and weight-update steps; once BoA quantizes one weight, it updates the remaining (not-yet-quantized) weights to compensate for the task loss degradation caused by the quantization. The update formula to compensate for the quantization of the q-th weight w_q is formulated as (Frantar et al., 2023)

$$\boldsymbol{\delta}\mathbf{w} = \frac{\mathcal{Q}(w_q) - w_q}{[\mathbf{U}]_{q,q}} [\mathbf{U}]_{q,:} \text{ where } \mathbf{U} = \text{Chol}(\mathbf{H}^{-1})^T, \tag{2}$$

where $\mathbf{H}$ is the Hessian and $\text{Chol}(\cdot)$ denotes a Cholesky decomposition (*i.e.*, $\mathbf{U}$ is an upper triangular matrix satisfying

Table 1. Approximated Hessians in GPTQ and the proposed BoA

Method	Layer	$\mathbf{H} = \mathbf{H}_{\text{col}} \otimes \mathbf{H}_{\text{row}}$
GPTQ	$\mathbf{W}_{\{Q,K,V\}}$	$2\mathbf{X}\mathbf{X}^T \otimes \mathbf{I}$
BoA	$\mathbf{W}_{Q,h}$	$2\mathbf{X}\mathbf{X}^T \otimes \mathbf{K}_h^T\mathbf{K}_h$
	$\mathbf{W}_{K,h}$	$2\mathbf{X}\mathbf{X}^T \otimes \mathbf{Q}_h^T\mathbf{Q}_h$
	$\mathbf{W}_{V,h}$	$2\mathbf{X}\mathbf{A}_h^T\mathbf{A}_h\mathbf{X}^T \otimes \mathbf{W}_{\text{out},h}^T\mathbf{W}_{\text{out},h}$
	$\mathbf{W}_{\text{out},h}$	$2\mathbf{X}_{\text{out},h}\mathbf{X}_{\text{out},h}^T \otimes \mathbf{I}$

* For $\mathbf{W}_{Q,h}$ and $\mathbf{W}_{K,h}$ of models exploiting rotary position embedding, see (14) and (15).

$\mathbf{H}^{-1} = \mathbf{U}^T\mathbf{U}$).

The key difference over GPTQ lies in the approximation of the Hessian $\mathbf{H}$. In GPTQ, the layer-wise reconstruction error $\|\Delta\mathbf{W}^{(\ell)}\mathbf{X}^{(\ell-1)}\|_F^2$ in (1) has been used (*i.e.*, the layer-wise independence has been assumed) to approximate $\mathbf{H}$ which yields the following Hessian equation[2]:

$$\mathbf{H}^{(\mathbf{w}^{(\ell)})} \approx 2\mathbf{X}^{(\ell-1)}\mathbf{X}^{(\ell-1)^T} \otimes \mathbf{I}, \tag{3}$$

where $\mathbf{w}^{(\ell)}$ is the flattened representation of $\mathbf{W}^{(\ell)}$, $\mathbf{H}^{(\mathbf{w}^{(\ell)})}$ is the Hessian for the ℓ-th layer, $\otimes$ denotes the Kronecker product operation, and $\mathbf{I}$ is the identity matrix. The approximated Hessian $\mathbf{H}^{(\mathbf{w}^{(\ell)})}$ in GPTQ relies solely on the input, which means that GPTQ cannot consider the influence of other layers. In other words, GPTQ neglects inter-layer dependencies within the attention module, a crucial aspect of Transformers, which results in limited performance (Jeon et al., 2023b). To overcome this, we develop Hessians that incorporate inter-layer dependencies and then use them instead of the conventional Hessian in (3).

In Table 1, we summarize the proposed Hessians; the detailed derivation is provided in the following subsections. It can be observed that the proposed Hessians not only contain the term related to the input $\mathbf{X}$, but also involve the terms related to other layers (*e.g.*, $\mathbf{K}_h^T\mathbf{K}_h$ for $\mathbf{W}_Q$).

3.2. Proposed Attention-aware Hessian

To consider the inter-layer dependencies within the attention module, we exploit the attention reconstruction error rather than the layer-wise reconstruction error when approximating the Hessian. For an input sequence $\mathbf{X} \in \mathbb{R}^{d\times L}$, the output of the multi-head attention (MHA) is expressed as

$$\text{MHA}(\mathbf{X}) = \sum_{h=1}^{H} \mathbf{W}_{\text{out},h}(\mathbf{A}_h\mathbf{V}_h)^T, \mathbf{A}_h = \sigma\left(\frac{\mathbf{Q}_h\mathbf{K}_h^T}{\sqrt{d_h}}\right), \tag{4}$$

where $\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h \in \mathbb{R}^{L\times d_h}$ are query, key, value for the h-th attention head, d_h is the head dimension, σ is the row-wise softmax function, and H is the total number of heads.

[2]The second-order derivative of $\|\mathbf{M}_1\Delta\mathbf{W}\mathbf{M}_2\|_F^2$ with respect to $\Delta\mathbf{w}$ is $2\mathbf{M}_2\mathbf{M}_2^T \otimes \mathbf{M}_1^T\mathbf{M}_1$ (see Appendix A for the proof).

Hessian for query When quantizing the query projection weights $\mathbf{W}_{Q,h}$, $\mathbf{W}_{\text{out},h}$ and $\mathbf{V}_h$ remain unchanged, but the attention weights $\mathbf{A}_h$ change. Using the first-order Taylor polynomial, the perturbation in $\mathbf{A}_h$ can be approximated as

$$\begin{aligned}\Delta\mathbf{A}_h &= \sigma\left(\frac{(\mathbf{Q}_h + \Delta\mathbf{Q}_h)\mathbf{K}_h^T}{\sqrt{d_h}}\right) - \sigma\left(\frac{\mathbf{Q}_h\mathbf{K}_h^T}{\sqrt{d_h}}\right) \\ &\approx \frac{\Delta\mathbf{Q}_h\mathbf{K}_h^T}{\sqrt{d_h}}\mathbf{J}_\sigma^T = \frac{\mathbf{X}^T\Delta\mathbf{W}_{Q,h}^T\mathbf{K}_h^T\mathbf{J}_\sigma^T}{\sqrt{d_h}},\end{aligned} \tag{5}$$

where $\mathbf{J}_\sigma$ is the Jacobian matrix of the softmax function σ. Thus, the attention reconstruction error is expressed as

$$\begin{aligned}\|\Delta\text{MHA}(\mathbf{X})\|_F^2 &= \|\mathbf{W}_{\text{out},h}(\Delta\mathbf{A}_h\mathbf{V}_h)^T\|_F^2 \\ &\approx \left\|\frac{\mathbf{W}_{\text{out},h}\mathbf{V}_h^T\mathbf{J}_\sigma\mathbf{K}_h}{\sqrt{d_h}}\Delta\mathbf{W}_{Q,h}\mathbf{X}\right\|_F^2,\end{aligned} \tag{6}$$

which yields the following Hessian for $\mathbf{W}_{Q,h}$ (see Footnote 2):

$$\mathbf{H}^{(\mathbf{w}_{Q,h})} = 2\mathbf{X}\mathbf{X}^T \otimes \frac{\mathbf{K}_h^T\mathbf{J}_\sigma^T\mathbf{V}_h\mathbf{W}_{\text{out},h}^T\mathbf{W}_{\text{out},h}\mathbf{V}_h^T\mathbf{J}_\sigma\mathbf{K}_h}{d_h}. \tag{7}$$

Hessian for key As in the quantization of the query projection weights, the attention weight $\mathbf{A}_h$ changes when quantizing the key projection weights $\mathbf{W}_{K,h}$. By following the steps for (5), $\mathbf{A}_h$ can be approximated as

$$\Delta\mathbf{A}_h \approx \frac{\mathbf{Q}_h\Delta\mathbf{K}_h^T}{\sqrt{d_h}}\mathbf{J}_\sigma^T = \frac{\mathbf{Q}_h\Delta\mathbf{W}_{K,h}\mathbf{X}\mathbf{J}_\sigma^T}{\sqrt{d_h}},$$

and then the attention reconstruction error is expressed as

$$\|\Delta\text{MHA}(\mathbf{X})\|_F^2 \approx \left\|\frac{\mathbf{Q}_h}{\sqrt{d_h}}\Delta\mathbf{W}_{K,h}\mathbf{X}\mathbf{J}_\sigma^T\mathbf{V}_h\mathbf{W}_{\text{out},h}^T\right\|_F^2.$$

Thus, we obtain the following Hessian for $\mathbf{W}_{K,h}$:

$$\mathbf{H}^{(\mathbf{w}_{K,h})} = 2\mathbf{X}\mathbf{J}_\sigma^T\mathbf{V}_h\mathbf{W}_{\text{out},h}^T\mathbf{W}_{\text{out},h}\mathbf{V}_h^T\mathbf{J}_\sigma\mathbf{X}^T \otimes \frac{\mathbf{Q}_h^T\mathbf{Q}_h}{d_h}. \tag{8}$$

Hessian for value When quantizing the weights $\mathbf{W}_{V,h}$ of the value projection, only $\mathbf{V}_h$ changes. Thus, we have

$$\begin{aligned}\|\Delta\,\text{MHA}(\mathbf{X})\|_F^2 &= \|\mathbf{W}_{\text{out},h}(\mathbf{A}_h\Delta\mathbf{V}_h)^T\|_F^2 \\ &= \|\mathbf{W}_{\text{out},h}\Delta\mathbf{W}_{V,h}\mathbf{X}\mathbf{A}_h^T\|_F^2,\end{aligned}$$

which yields the following Hessian for $\mathbf{W}_{V,h}$:

$$\mathbf{H}^{(\mathbf{w}_{V,h})} = 2\mathbf{X}\mathbf{A}_h^T\mathbf{A}_h\mathbf{X}^T \otimes \mathbf{W}_{\text{out},h}^T\mathbf{W}_{\text{out},h}. \tag{9}$$

Hessian for out When the out-projection weights $\mathbf{W}_{\text{out},h}$ are quantized, the attention reconstruction error is

$$\|\Delta\,\text{MHA}(\mathbf{X})\|_F^2 = \|\Delta\mathbf{W}_{\text{out},h}(\mathbf{A}_h\mathbf{V}_h)^T\|_F^2.$$

Thus, the corresponding Hessian is

$$\mathbf{H}^{(\mathbf{w}_{out,h})} = 2\mathbf{V}_h^T\mathbf{A}_h^T\mathbf{A}_h\mathbf{V}_h \otimes \mathbf{I} = 2\mathbf{X}_{\text{out},h}\mathbf{X}_{\text{out},h}^T \otimes \mathbf{I}, \tag{10}$$

where $\mathbf{X}_{\text{out},h} = (\mathbf{A}_h\mathbf{V}_h)^T$.

Algorithm 1 BoA

Input: weights $\mathbf{W}_{\{Q,K,V\}} \in \mathbb{R}^{H\times d_h\times d}$ and inputs $\mathbf{X}$ of the Transformer layer
for $\mathbf{W} \in \{\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V\}$ **do**
  Initialize quantized output: $\mathbf{Q} \leftarrow \mathbf{0}_{H\times d_h\times d}$
  Initialize (row-wise) quantization errors: $\mathbf{E} \leftarrow \mathbf{0}_{H\times d}$
  Compute attention-aware Hessians: $\mathbf{H}_h = \mathbf{H}_{\text{col},h} \otimes \mathbf{H}_{\text{row},h}$ ▷ see Table 1
  Set step size (scale) $\mathbf{S}$: $\min_{\mathbf{S}} \text{tr}\big(\Delta\mathbf{W}_h\mathbf{H}_{\text{col},h}\Delta\mathbf{W}_h^T\big)$
  Compute inverse Hessians $\mathbf{H}_{\text{col},h}^{-1}$ and $\mathbf{H}_{\text{row},h}^{-1}$
  Compute $\mathbf{U}_{\text{col},h} = \text{Chol}(\mathbf{H}_{\text{col},h}^{-1})^T$ and $\mathbf{U}_{\text{row},h} = \text{Chol}(\mathbf{H}_{\text{row},h}^{-1})^T$
  for $j = 0, \dots, d_h - 1$ **do**
    Construct $\mathbf{W}^{(j)} \in \mathbb{R}^{H\times d}$ by stacking the j-th rows $[\mathbf{W}_h]_{j,:}$
    Quantize $\mathbf{W}^{(j)}$: $(\mathbf{Q}_{:,j,:}, \mathbf{E}) \leftarrow \text{GPTQ}(\mathbf{W}^{(j)}, \mathbf{U}_{\text{col},h}, \mathbf{S})$ ▷ see Appendix E
    Update remaining rows: $[\mathbf{W}_h]_{j:,:} \leftarrow [\mathbf{W}_h]_{j:,:} - \frac{[\mathbf{U}_{\text{row},h}^T]_{j:,j}\cdot\mathbf{E}_{h,:}\cdot\mathbf{U}_{\text{col},h}}{[\mathbf{U}_{\text{row},h}]_{j,j}}$ ▷ see Proposition 3.1
  end for
end for
Output: quantized weights $\mathbf{Q}$

3.3. Efficient Implementation of BoA

While inter-layer dependencies within the attention module can be considered by exploiting the proposed Hessians, they are significantly more complex than the conventional Hessian in (3), which may incur high computational costs. For example, computing the proposed Hessians in (7) and (8) would be more expensive than computing the conventional one in (3). In this subsection, we present techniques to mitigate the computational overheads incurred by the proposed attention-aware Hessians.

Hessian relaxation The largest overhead related to the computation of the proposed Hessians is the Jacobian matrix $\mathbf{J}_\sigma$ in (7) and (8). For an input sequence of length L, the shape of $\mathbf{J}_\sigma$ is $H\times L\times L\times L$, which requires a large amount of memory and high computational cost (*e.g.*, more than 400 GB even for the OPT-125M model when $L = 2048$).

To mitigate such overhead, we establish a relaxed Hessian that does not require computing $\mathbf{J}_\sigma$. To this end, we build the following upper bound for the attention reconstruction error in (6), which will be used as its surrogate:

$$\|\Delta\,\text{MHA}(\mathbf{X})\|_F^2 \le \left\|\frac{\mathbf{W}_{\text{out},h}\mathbf{V}_h^T\mathbf{J}_\sigma}{\sqrt{d_h}}\right\|_F^2 \cdot \|\mathbf{K}_h\Delta\mathbf{W}_{Q,h}\mathbf{X}\|_F^2\,.$$

Noting that the constant term $\|\mathbf{W}_{\text{out},h}\mathbf{V}_h^T\mathbf{J}_\sigma\|_F^2$ does not affect quantization,[3] we use the term $\|\mathbf{K}_h\Delta\mathbf{W}_{Q,h}\mathbf{X}\|_F^2$ as a surrogate of the attention reconstruction error when deriving the Hessian for $\mathbf{W}_{Q,h}$, which yields the following Hessian:

$$\mathbf{H}^{(\mathbf{w}_{Q,h})} = 2\mathbf{X}\mathbf{X}^T \otimes \mathbf{K}_h^T\mathbf{K}_h. \tag{11}$$

Similarly, we can establish a relaxed Hessian for $\mathbf{W}_{K,h}$ as

$$\mathbf{H}^{(\mathbf{w}_{K,h})} = 2\mathbf{X}\mathbf{X}^T \otimes \mathbf{Q}_h^T\mathbf{Q}_h. \tag{12}$$

[3]The update $\boldsymbol{\delta}\mathbf{w}$ in (2) is not affected by the constant multiple of $\mathbf{H}$ because $[c\mathbf{U}]_{q,:}/[c\mathbf{U}]_{q,q} = [\mathbf{U}]_{q,:}/[\mathbf{U}]_{q,q}$ for any constant c.

We note that if rotary position embedding (RoPE) (Su et al., 2023) is used, the MHA output is different from that in (4), and thus the corresponding attention-aware Hessians would also be different. Specifically, since the attention weight $\mathbf{A}_h$ for models exploiting RoPE is expressed as

$$\mathbf{A}_h = \sigma\left(\frac{\widetilde{\mathbf{Q}}_h\widetilde{\mathbf{K}}_h^T}{\sqrt{d_h}}\right), \widetilde{\mathbf{Q}}_h = \text{RoPE}(\mathbf{Q}_h), \widetilde{\mathbf{K}}_h = \text{RoPE}(\mathbf{K}_h), \tag{13}$$

the surrogate $\|\mathbf{K}_h\Delta\mathbf{Q}_h^T\|_F^2$ used to develop the attention-aware Hessian in (11) changes to $\|\widetilde{\mathbf{K}}_h\Delta\widetilde{\mathbf{Q}}_h^T\|_F^2$. As a result, we obtain different Hessians for models exploiting RoPE (see Appendix B for the detailed derivation):

$$\mathbf{H}^{(\mathbf{w}_{Q,h})} = 2\mathbf{X}\mathbf{X}^T \otimes \frac{1}{L}\sum_{\ell=1}^{L}\mathbf{R}_\ell^T\widetilde{\mathbf{K}}_h^T\widetilde{\mathbf{K}}_h\mathbf{R}_\ell, \tag{14}$$

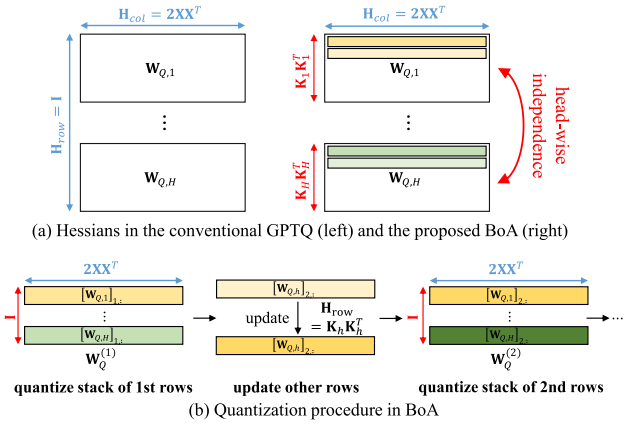
$$\mathbf{H}^{(\mathbf{w}_{K,h})} = 2\mathbf{X}\mathbf{X}^T \otimes \frac{1}{L}\sum_{\ell=1}^{L}\mathbf{R}_\ell^T\widetilde{\mathbf{Q}}_h^T\widetilde{\mathbf{Q}}_h\mathbf{R}_\ell, \tag{15}$$

where $\mathbf{R}_\ell$ be the rotary matrix for the ℓ-th token (see eq. (15) in (Su et al., 2023)).

We summarize the relaxed Hessians in Table 1.

Efficient computation of inverse Hessians After computing Hessians, their inverse matrices need to be computed to update the remaining weights after each quantization (see (2)). Owing to the size of the proposed Hessians being $dd_h \times dd_h$, the complexity of the computation of the inverse Hessian would be $\mathcal{O}(d^3d_h^3)$ in our approach. This is considerably more expensive than the complexity $\mathcal{O}(d^3)$ of GPTQ, where the inverse of only $\mathbf{X}\mathbf{X}^T \in \mathbb{R}^{d\times d}$ in (3) is needed (Frantar et al., 2023).

For the efficient inverse computation, we exploit the useful properties of the Kronecker product (see (17a)-(17c) in



(a) Hessians in the conventional GPTQ (left) and the proposed BoA (right)

(b) Quantization procedure in BoA

Figure 1. Illustration of BoA for the query projection $\mathbf{W}_Q$.

Appendix A). Specifically, let $\mathbf{H} = \mathbf{H}_{\text{col}} \otimes \mathbf{H}_{\text{row}}$ where $\mathbf{H}_{\text{col}} \in \mathbb{R}^{d\times d}$ and $\mathbf{H}_{\text{row}} \in \mathbb{R}^{d_h\times d_h}$, then we obtain

$$\mathbf{H}^{-1} = (\mathbf{H}_{\text{col}} \otimes \mathbf{H}_{\text{row}})^{-1} = \mathbf{H}_{\text{col}}^{-1} \otimes \mathbf{H}_{\text{row}}^{-1}.$$

This implies that the inverse Hessian $\mathbf{H}^{-1}$ can be obtained by computing $\mathbf{H}_{\text{col}}^{-1}$ and $\mathbf{H}_{\text{row}}^{-1}$ (line 5 in Algorithm 1) whose complexity is $\mathcal{O}(d^3) + \mathcal{O}(d_h^3)$ $(= \mathcal{O}(d^3))$, not $\mathcal{O}(d^3 d_h^3)$. Similarly, we can efficiently perform the Cholesky decomposition (*i.e.*, $\text{Chol}(\mathbf{H}^{-1})$ in (2)) with the same order of complexity as in GPTQ. Specifically, if $\mathbf{L}_1 = \text{Chol}(\mathbf{H}_{\text{col}}^{-1})$ and $\mathbf{L}_2 = \text{Chol}(\mathbf{H}_{\text{row}}^{-1})$, $\mathbf{H}^{-1}$ can be expressed as

$$\mathbf{H}^{-1} = \mathbf{L}_1\mathbf{L}_1^T \otimes \mathbf{L}_2\mathbf{L}_2^T = (\mathbf{L}_1 \otimes \mathbf{L}_2)(\mathbf{L}_1 \otimes \mathbf{L}_2)^T.$$

Subsequently, noting that the Kronecker product of lower triangular matrices is also lower triangular, we obtain

$$\text{Chol}(\mathbf{H}^{-1}) = \mathbf{L}_1 \otimes \mathbf{L}_2 = \text{Chol}(\mathbf{H}_{\text{col}}^{-1}) \otimes \text{Chol}(\mathbf{H}_{\text{row}}^{-1}).$$

Thus, we can obtain $\text{Chol}(\mathbf{H}^{-1})$ by computing $\text{Chol}(\mathbf{H}_{\text{col}}^{-1})$ and $\text{Chol}(\mathbf{H}_{\text{row}}^{-1})$ (line 6 in Algorithm 1). Consequently, the computational complexity of the Choleksy decomposition would be $\mathcal{O}(d^3)$, not $\mathcal{O}(d^3 d_h^3)$.

Simultaneous quantization of different heads Because the proposed Hessians model the dependency between different rows (*e.g.*, $\mathbf{H}_{\text{row}} = \mathbf{K}_h^T\mathbf{K}_h$ for $\mathbf{W}_{Q,h}$; see (11)), we can compensate for the quantization error of a certain row by updating other rows. Specifically, the row-update formula is formulated as in the following proposition for given Hessian $\mathbf{H} = \mathbf{H}_{\text{col}} \otimes \mathbf{H}_{\text{row}}$.

Proposition 3.1. *Let $\mathbf{W}_h$ be a $d_h \times d$ matrix whose Hessian is $\mathbf{H}_h = \mathbf{H}_{col,h} \otimes \mathbf{H}_{row,h}$. Suppose $\mathbf{W}_h$ is quantized via (2). If the j-th row of $\mathbf{W}_h$ has been quantized, then the update formula to compensate for the quantization is expressed as*

$$[\delta\mathbf{W}_h]_{j:,:} = -\frac{[\mathbf{U}_{row,h}^T]_{j:,j} \cdot \mathbf{e} \cdot \mathbf{U}_{col,h}}{[\mathbf{U}_{row,h}]_{j,j}}, \tag{16}$$

Table 2. Processing time (hour) of BoA with and without simultaneous quantization of different heads

Simultaneous Quantization	LLaMA Model Size		
	7B	13B	30B
X	27.75	51.66	135.4
O	**0.961**	**1.553**	**3.295**

where $\mathbf{U}_{col,h} = \text{Chol}(\mathbf{H}_{col,h}^{-1})^T$, $\mathbf{U}_{row,h} = \text{Chol}(\mathbf{H}_{row,h}^{-1})^T$, and $\mathbf{e} \in \mathbb{R}^{1\times d}$ is the quantization error of the j-th row defined as $e_i = ([\mathbf{W}_h]_{j,i} - \mathcal{Q}([\mathbf{W}_h]_{j,i}))/[\mathbf{U}_{col,h}]_{i,i}$.

Proof. See Appendix C. ■

The update formula in (16) implies that we do not need to compute and store the Cholesky decomposition $\mathbf{U}_h = \mathbf{U}_{\text{col},h} \otimes \mathbf{U}_{\text{row},h}$ of the full Hessian $\mathbf{H}_h$ for updating weights; only $\mathbf{U}_{\text{col},h}$ and $\mathbf{U}_{\text{row},h}$ are sufficient for updating weights, and thus we can save the memory cost caused by the high-dimensional Kronecker product operation. Proposition 3.1 also implies that the conventional GPTQ cannot compensate quantization error of certain row by updating other rows because $\mathbf{H}_{\text{row}} = \mathbf{I}$ (see (3)) and thus $\mathbf{U}_{\text{row}} = \mathbf{I}$ and $[\delta\mathbf{W}]_{i,:} = \mathbf{0}$ for all $i \neq j$.

While the proposed BoA can compensate for the quantization error of each row, the rows must be quantized sequentially (not simultaneously). For example, the second row can be quantized after being updated to compensate for the quantization error of the first row. To accelerate the quantization process, we assume independence between different attention heads (see Figure 1(a)), under which rows related to different heads are independent and can thus be quantized together. For a better understanding, we consider the query projection $\mathbf{W}_Q$ as an example (see Figure 1(b)). In the quantization step, we stack the j-th rows $[\mathbf{W}_{Q,h}]_{j,:}$ of all different heads, constructing the sub-weight matrix $\mathbf{W}_Q^{(j)} \in \mathbb{R}^{H\times d}$ (line 8 in Algorithm 1). Because the rows of $\mathbf{W}_Q^{(j)}$ are mutually independent, all the rows of $\mathbf{W}_Q^{(j)}$ can be quantized simultaneously as in GPTQ (line 9 in Algorithm 1). Following the quantization of j-th rows, we compensate for the quantization error by updating the remaining rows. In this update step, we use the refined weight-update formula in (16) (line 10 in Algorithm 1).

We measure quantization processing times of BoA with and without simultaneous quantization to evaluate how much the head-wise joint quantization can accelerate the quantization process. From Table 2, we observe that BoA requires a significantly long processing time without the simultaneous quantization (more than one day even for the 7B model). This is because all rows need to be quantized sequentially (*e.g.*, 4096 rows are quantized sequentially for LLaMA-7B) and thus the massive compute capabilities of modern GPUs cannot be utilized properly. As evident, we can significantly

Table 3. Weight-only quantization performance on LLaMA2 and LLaMA3 models without transformation

Model	Precision	Method	Wiki2 PPL (↓)	Zero-shot Accuracy (↑)								
				Arc-c	Arc-e	BQ	HS	LAMB	OBQA	PIQA	WG	Average
LLaMA2-7B	FP16	Baseline	5.473	45.90	74.66	77.92	75.94	70.86	44.00	78.89	68.90	67.13
	INT2	RTN	7.8e3	26.45	26.18	39.30	25.99	0.00	24.20	49.40	49.96	30.19
		GPTQ	30.85	26.96	44.15	56.30	42.52	23.95	28.20	63.22	54.70	42.50
		BoA	**12.76**	28.33	47.73	64.71	45.46	31.33	29.60	64.85	54.46	**45.81**
	INT3	RTN	342.4	22.53	35.06	44.13	28.87	1.63	26.40	58.11	48.70	33.18
		GPTQ	6.719	36.95	59.72	65.54	66.68	58.21	39.20	74.65	66.06	58.38
		BoA	**6.007**	40.27	69.28	74.22	72.00	66.60	41.20	78.35	67.64	**63.70**
LLaMA2-13B	FP16	Baseline	4.885	49.06	77.65	80.49	79.38	73.41	45.80	80.69	72.22	69.84
	INT2	RTN	5.7e3	27.47	26.89	37.95	25.97	0.00	25.20	49.08	48.38	30.12
		GPTQ	35.08	21.76	35.31	61.96	35.22	19.41	28.80	57.29	52.88	39.08
		BoA	**18.33**	29.78	46.30	62.23	49.30	29.33	27.60	63.33	52.64	**45.06**
	INT3	RTN	227.2	23.98	28.75	49.24	28.51	4.46	24.40	53.26	50.75	32.92
		GPTQ	9.790	34.81	62.12	67.06	55.57	47.61	37.00	73.18	61.33	54.84
		BoA	**5.833**	43.52	69.28	78.93	74.71	65.22	35.20	77.42	62.51	**63.35**
LLaMA3-8B	FP16	Baseline	6.137	53.67	77.61	81.19	79.15	72.23	45.00	81.01	73.24	70.39
	INT2	RTN	6.6e4	25.51	25.80	53.94	26.34	0.00	29.00	51.52	50.20	32.79
		GPTQ	24.54	23.29	32.58	53.39	39.17	7.10	27.00	53.32	52.25	36.01
		BoA	**21.70**	26.62	44.87	60.52	43.34	16.89	29.40	59.85	55.80	**42.16**
	INT3	RTN	129.1	23.21	33.88	55.60	34.83	4.60	25.20	58.22	52.57	36.01
		GPTQ	8.226	41.47	63.01	75.47	70.05	59.53	40.60	73.78	69.85	61.72
		BoA	**7.782**	45.14	72.77	78.69	72.66	62.41	42.60	77.31	71.35	**65.37**
LLaMA3.2-1B	FP16	Baseline	13.15	38.14	63.26	69.51	60.78	54.38	34.60	74.37	59.51	56.82
	INT2	RTN	6.3e4	26.96	25.59	41.53	26.05	0.01	26.40	51.52	50.59	31.08
		GPTQ	538.9	25.26	26.64	37.83	26.41	0.22	27.60	51.41	48.46	30.48
		BoA	**312.2**	25.09	26.85	40.06	27.17	1.42	27.00	51.96	51.07	**31.33**
	INT3	RTN	1.9e3	25.60	26.94	54.13	29.90	0.58	27.00	52.18	49.17	33.19
		GPTQ	112.0	24.06	39.48	53.85	31.07	13.85	27.80	60.07	49.33	37.44
		BoA	**26.43**	30.63	55.26	59.97	48.33	31.95	29.60	66.65	53.99	**47.05**
LLaMA3.2-3B	FP16	Baseline	11.04	46.16	67.80	78.62	70.44	62.15	36.00	75.52	67.40	63.01
	INT2	RTN	2.0e4	26.79	26.52	37.89	25.93	0.00	30.80	50.92	49.09	30.99
		GPTQ	98.19	24.83	27.78	52.32	33.83	4.38	28.60	52.23	51.14	34.39
		BoA	**54.64**	25.77	35.48	57.52	35.63	14.42	29.00	56.96	53.43	**38.53**
	INT3	RTN	882.6	26.37	27.86	45.87	37.04	1.44	26.00	53.92	48.46	33.37
		GPTQ	46.14	28.92	37.63	44.01	39.27	18.76	28.60	61.64	54.70	39.19
		BoA	**13.64**	42.32	66.12	77.52	64.46	54.18	35.20	72.69	62.51	**59.38**

* Results for high bit-widths and results on LLaMA1 and OPT models are provided in Appendix D.1 due to the page limitation.

reduce the processing time by applying the head-wise joint quantization (more than 40 times reduction on the 30B model).

4. Experiments

In this section, we evaluate the weight-only quantization performance of the proposed BoA (Section 4.2) and synergy with transformation-based methods in terms of weight-only and weight-activation quantization (Sections 4.2 and 4.3).

4.1. Experimental Setup

We conduct experiments on OPT (Zhang et al., 2022), LLaMA (Touvron et al., 2023a)), LLaMA2 (Touvron et al., 2023b), and LLaMA3. As in previous studies (Shao et al., 2023; Ma et al., 2024; Lin et al., 2024; Ashkboos et al., 2024; Liu et al., 2024), we construct a calibration dataset by sampling 128 random sequences of length 2048 from WikiText-2 (Merity et al., 2016). As a performance metric, we use the perplexity (PPL) score on the WikiText-2 test dataset and accuracy on eight zero-shot commonsense reasoning tasks (ARC-challenge (Arc-c) and ARC-easy (Arc-e) (Clark et al., 2018), BoolQ (BQ) (Clark et al., 2019), HellaSwag (HS) (Zellers et al., 2019), LAMBADA (LAMB) (Paperno et al., 2016), OpenbookQA (OBQA) (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), and WinoGrande (WG) (Sakaguchi et al., 2021)). All experiments were conducted using a single NVIDIA H100 GPU (80 GB).

When determining a quantization order in BoA, the heuristic introduced by GPTQ can be used; the column/row corresponding to the largest $\mathrm{diag}(\mathbf{H}_{\mathrm{col}})/\mathrm{diag}(\mathbf{H}_{\mathrm{row}})$ (*i.e.*, the most quantization-sensitive column/row) is first quantized for better compensation. Empirically, we observed that this heuristic could occasionally enhance the performance, yet at other times, it may result in inferior performance. For GPTQ and the proposed BoA, we conduct experiments with and without this heuristic and report the better results.

Table 4. Memory costs of GPTQ and the proposed BoA

Method	Memory Cost (GB)			Wiki2 PPL (↓)		
	7B	13B	30B	7B	13B	30B
GPTQ	4.426	5.872	8.271	19.63	34.63	9.770
BoA	9.550	16.56	32.79	10.28	8.309	6.669
Relaxed BoA*	6.446	8.397	11.55	10.53	8.700	7.007

* Attention-aware Hessians have been applied to query and key, but not to value.

Table 5. Processing time (hour) of different quantization methods

Method	Inter-layer Dependency	Optimization Type	LLaMA Model Size		
			7B	13B	30B
GPTQ	X	one-shot	0.12	0.20	0.43
OmniQuant	O	gradient -based	1.83	3.32	7.66
AffineQuant			4.32	8.41	21.4
DuQuant+LWC			1.22	2.08	4.55
BoA	O	one-shot	**0.96**	**1.55**	**3.30**

4.2. Weight-Only Quantization Results

Comparison with GPTQ We first compare the proposed BoA with GPTQ to demonstrate the efficacy of the proposed attention-aware Hessians. In this experiment, we do not apply any model transformation method (*e.g.* QuaRot (Ashkboos et al., 2024) and SpinQuant (Liu et al., 2024)) to solely evaluate the effectiveness of the proposed Hessians.

In Table 3 and Tables 8-10 (see Appendix D.1), we summarize the PPL and zero-shot task performances of BoA and GPTQ. We also include the performance of the rounding-to-nearest (RTN) method which naively assigns the nearest quantization bin. We observe BoA and GPTQ exhibit reasonable PPL even for INT2 quantization where RTN collapses significantly (*i.e.*, PPL is almost 10^4). This is because BoA and GPTQ minimize the task loss degradation rather than the weight quantization error $\Delta\mathbf{W}$. As evident, BoA significantly surpasses GPTQ on all models in both PPL and zero-shot accuracy. For example, BoA achieves 10%p accuracy improvement on the 3-bit quantized LLaMA2-13B and LLaMA3.2-1B models. In addition, BoA shows 20%p higher accuracy for the 3-bit quantized LLaMA3.2-3B model. The key factor leading to such an outstanding performance is that BoA considers inter-layer dependencies within the attention module by exploiting the proposed attention-aware Hessians while GPTQ assumes the independence of layers.

In Table 4, we summarize the memory costs of BoA and GPTQ. We observe that BoA requires larger memory because BoA additionally uses outputs of other layers to consider inter-layer dependencies. It is worth mentioning that when the memory resource is limited, BoA can still be used with a slight relaxation. Noting that the large memory cost of BoA is attributable to the Hessian for the value projection ($\mathbf{H}_{\text{col},h} = 2\mathbf{X}\mathbf{A}_h^T\mathbf{A}_h\mathbf{X}^T$ in (9)) whose shape is $H \times d \times d$, we can greatly reduce the memory cost by exploiting the standard Hessian in (3) for the value projection and applying the proposed attention-aware Hessians only for query and key projections. In doing so, BoA needs slightly more memory (*e.g.* 3 GB for 30B; see Table 4) yet still performs much better than GPTQ.

We now compare the processing times of BoA and GPTQ. As evident from Table 5, BoA requires a longer processing time than that needed by GPTQ. This is because GPTQ quantizes all the rows simultaneously, while BoA sequentially quantizes them to compensate for the quantization error of each row (see Figure 1(b)). Clearly, there exists a trade-off between quantization speed and accuracy. In real cases, when one needs to consider inter-layer dependencies to preserve the performance of the original model as much as possible, the proposed BoA would be an intriguing solution when compared to existing gradient-based approaches (see Table 5 and Table 6). When faster quantization is required, one possible solution is to reduce the number of sequential quantizations by quantizing multiple rows in each head simultaneously, which will be considered in our future studies.

Comparison with transformation-based methods To improve the quantization performance, recent studies have transformed models by applying smoothing, rotation, or permutation (Shao et al., 2023; Ma et al., 2024; Ashkboos et al., 2024; Liu et al., 2024; Lin et al., 2024). In this experiment, we evaluate BoA under those model transformations (see Table 6 and Tables 11-12 in Appendix D.2). For conventional methods, we ran the official codes provided by the authors and reported the obtained results; details for implementing conventional methods are provided in Appendix D.2. When measuring the performance of BoA, we also transformed the model for a fair comparison. We applied QuaRot for the transformation because QuaRot does not require any training and incurs no extra costs during the actual inference (Ashkboos et al., 2024).

We observe that the performance of the proposed BoA is boosted significantly when applying the transformation. For example, the PPLs of the 2-bit quantized LLaMA2-13B / LLaMA3-8B models improve from 18.33 / 21.70 to 8.237 / 15.24, respectively (see Tables 3 and 6). For INT3 quantization, BoA almost preserves the performance of the original full-precision model; the accuracy drop is 2.3%p for LLaMA3-8B and 1.3%p for LLaMA2-13B (see Table 11). For all models, we achieve at least 5%p improvement in the zero-shot accuracy (in particular 11%p improvement for the 2-bit quantized LLaMA2-13B model). We note that the performance gap between BoA and GPTQ remains significant after the model transformation; for example, the zero-shot accuracy of BoA on the 2-bit quantized LLaMA2-7B model is 12%p larger than that obtained by QuaRot-GPTQ.

Finally, we emphasize that BoA not only surpasses existing

Table 6. Weight-only quantization performance on transformed LLaMA2 and LLaMA3 models

Model	Precision	Method	Wiki2 PPL (↓)	Zero-shot Accuracy (↑)								
				Arc-c	Arc-e	BQ	HS	LAMB	OBQA	PIQA	WG	Average
LLaMA2-7B	FP	Baseline	5.473	45.90	74.66	77.92	75.94	70.86	44.00	78.89	68.90	67.13
	INT2	OmniQuant	21.85	25.17	36.91	61.80	38.38	5.80	28.40	57.40	49.49	37.92
		AffineQuant	91.19	23.46	29.17	40.03	27.44	0.16	22.20	54.13	51.78	31.05
		QuaRot-RTN	1.1e4	27.22	26.05	37.83	26.19	0.01	25.60	50.54	48.70	30.27
		QuaRot-GPTQ	22.05	22.27	36.24	61.99	33.54	20.63	28.60	56.75	51.78	38.98
		DuQuant	1.6e4	26.62	25.72	39.08	26.06	0.00	22.20	49.73	50.12	29.94
		DuQuant + LWC	46.27	23.12	28.96	44.04	29.60	2.41	27.40	53.70	49.17	32.30
		BoA†	**10.42**	29.69	54.84	64.37	52.03	46.54	33.60	67.52	59.43	**51.00**
LLaMA2-13B	FP	Baseline	4.885	49.06	77.65	80.49	79.38	73.41	45.80	80.69	72.22	69.84
	INT2	OmniQuant	12.92	26.71	46.00	63.64	51.19	16.93	31.40	64.64	52.64	44.14
		AffineQuant	9.415	26.96	49.83	63.15	52.40	26.58	33.60	64.47	53.83	46.35
		QuaRot-RTN	7.9e3	27.22	26.26	37.83	25.74	0.00	24.00	49.02	49.17	29.91
		QuaRot-GPTQ	9.593	31.66	56.52	63.15	48.81	36.93	32.60	66.05	60.38	49.51
		DuQuant	1.4e4	28.33	26.64	44.43	26.12	0.00	25.00	50.16	49.57	31.28
		DuQuant + LWC	10.40	28.50	46.51	63.70	52.79	25.24	30.80	65.13	54.14	45.85
		BoA†	**8.237**	35.49	63.97	71.28	58.36	56.25	35.40	71.82	62.75	**56.92**
LLaMA3-8B	FP	Baseline	6.137	53.67	77.61	81.19	79.15	72.23	45.00	81.01	73.24	70.39
	INT2	OmniQuant*	955.8	22.78	28.24	37.83	26.18	0.00	27.20	52.83	49.01	30.51
		AffineQuant*	1.1e3	23.46	27.31	37.86	26.04	0.00	26.80	52.12	51.22	30.60
		QuaRot-RTN	3.5e5	26.19	25.21	39.02	26.86	0.00	28.60	50.38	49.41	30.71
		QuaRot-GPTQ	18.28	28.33	46.68	64.10	45.04	29.49	29.20	62.08	55.25	45.02
		DuQuant	1.4e6	25.00	25.34	47.37	26.68	0.00	29.80	51.96	48.93	31.89
		DuQuant + LWC	2.6e4	24.40	26.68	38.26	25.30	0.00	29.20	49.67	52.01	30.69
		BoA†	**15.24**	30.38	54.42	68.17	49.17	39.89	34.20	65.94	60.14	**50.29**
LLaMA3.2-1B	FP	Baseline	13.15	38.14	63.26	69.51	60.78	54.38	34.60	74.37	59.51	56.82
	INT2	OmniQuant*	302.3	22.01	31.69	37.95	27.64	0.14	25.60	54.95	50.12	31.26
		AffineQuant*	268.3	22.44	31.61	37.83	27.75	0.12	26.00	54.30	49.88	31.24
		QuaRot-RTN	2.6e5	26.62	25.84	38.75	26.75	0.00	28.20	51.09	51.07	31.04
		QuaRot-GPTQ	54.28	22.18	34.01	56.39	31.90	14.65	24.80	56.91	50.59	36.43
		DuQuant	4.9e4	25.51	25.97	39.94	26.17	0.00	29.80	48.97	49.64	30.75
		DuQuant + LWC	9.3e3	28.16	25.38	37.89	25.27	0.00	26.60	50.49	49.57	30.42
		BoA†	**40.86**	24.06	39.77	58.20	34.62	16.79	26.20	57.07	52.64	**38.67**
LLaMA3.2-3B	FP	Baseline	11.04	46.16	67.80	78.62	70.44	62.15	36.00	75.52	67.40	63.01
	INT2	OmniQuant*	273.4	23.38	31.82	56.42	29.35	0.28	27.80	56.15	51.07	34.53
		AffineQuant*	282.2	22.35	32.87	47.28	29.44	0.25	28.20	56.26	48.38	33.13
		QuaRot-RTN	2.3e4	26.19	24.49	47.09	26.55	0.00	30.40	51.74	48.30	31.85
		QuaRot-GPTQ	52.18	23.98	36.95	60.92	34.68	19.24	27.40	57.67	52.49	39.17
		DuQuant	7.7e4	25.17	25.55	41.10	26.01	0.00	28.40	51.47	50.28	31.00
		DuQuant + LWC	770.9	23.63	26.64	38.20	26.32	0.03	26.20	51.74	51.70	30.56
		BoA†	**33.40**	27.30	45.66	65.87	40.86	24.57	29.00	61.37	56.27	**43.86**

† BoA has been applied after transforming the model via QuaRot.
* The learnable equivalent transformation (LET) option has been deactivated because this option does not support models exploiting grouped-query attention (GQA).
** Results for high bit-widths and results on LLaMA1 models are provided in Appendix D.2 due to the page limitation.

transformation-based methods exploiting the naive nearest rounding (*i.e.*, OmniQuant, AffineQuant, and DuQuant + LWC) but also is much faster than those methods relying on the gradient-based optimization (see Table 5). For example, on the LLaMA3-8B model, BoA achieves 20%p accuracy improvement for INT2 (see Table 6) and 13%p improvement for INT3 (see Table 11) over OmniQuant, AffineQuant, and DuQuant + LWC, yet reduces the quantization processing time of OmniQuant and AffineQuant more than twofold (see Table 5). Moreover, the degree of reduction increases with the model size; BoA requires 7 times shorter processing time than that needed by AffineQuant on LLaMA-30B.

4.3. Weight-Activation Quantization Results

We now evaluate the weight-activation quantization performance (see Table 7 and Tables 13-15 in Appendix D.3). As in previous studies (Shao et al., 2023; Ma et al., 2024; Lin et al., 2024; Ashkboos et al., 2024; Liu et al., 2024), we quantize input activations to all linear layers and KV caches with the Min-Max nearest-rounding quantizer where quantization parameters are determined dynamically for each token. We use the notation 'WxAyKVz' to denote the x-bit weight quantization, y-bit input activation quantization, and z-bit KV cache quantization. In this experiment, when measuring the performance of RTN, GPTQ, and the proposed BoA, we use SpinQuant (instead of QuaRot) for the model transformation. It is worth noting that while QuaRot uses Hadamard matrices (that are independent of data) for the rotation (Ashkboos et al., 2024), SpinQuant optimizes rotation matrices that make models more robust to the activation quantization and thus could outperform QuaRot (Liu et al., 2024). In our experiments, rotation matrices have been op-

Table 7. Weight-activation quantization performance on transformed LLaMA2 and LLaMA3 models

Model	Precision	Method	Wiki2 PPL (↓)	Zero-shot Accuracy (↑)								
				Arc-c	Arc-e	BQ	HS	LAMB	OBQA	PIQA	WG	Average
LLaMA2-7B	FP16	Baseline	5.473	45.90	74.66	77.92	75.94	70.86	44.00	78.89	68.90	67.13
	W2A4KV4	OmniQuant	1.0e5	26.88	26.30	39.02	25.57	0.00	25.60	48.42	50.51	30.29
		AffineQuant	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
		SpinQuant-RTN	23.23	25.51	34.01	62.20	32.09	14.48	26.00	55.17	50.91	37.55
		SpinQuant-GPTQ	24.29	22.95	36.74	59.88	32.83	13.81	26.20	56.64	51.30	37.54
		DuQuant	6.4e3	28.41	26.73	38.01	25.65	0.00	26.00	48.48	51.62	30.61
		DuQuant + LWC	16.35	24.91	37.33	61.99	41.46	10.54	28.20	58.71	53.28	39.55
		BoA†	**11.80**	26.79	49.20	63.09	48.05	37.76	30.80	63.55	57.85	**47.14**
LLaMA2-13B	FP16	Baseline	4.885	49.06	77.65	80.49	79.38	73.41	45.80	80.69	72.22	69.84
	W2A4KV4	OmniQuant	3.8e3	26.88	26.30	37.83	25.48	0.00	23.60	48.20	49.96	29.78
		AffineQuant	1.2e3	26.62	27.19	37.83	26.37	0.00	24.60	48.80	49.17	30.07
		SpinQuant-RTN	11.71	26.96	40.61	63.12	44.54	29.64	29.20	60.88	53.67	43.58
		SpinQuant-GPTQ	15.54	23.55	41.29	62.17	35.76	16.50	29.80	59.52	51.54	40.02
		DuQuant	6.4e3	28.41	26.73	38.01	25.65	0.00	26.00	48.48	51.62	30.61
		DuQuant + LWC	16.35	24.91	37.33	61.99	41.46	10.54	28.20	58.71	53.28	39.55
		BoA†	**8.974**	31.74	55.98	64.80	56.22	49.18	34.40	68.44	59.27	**52.50**
LLaMA3-8B	FP16	Baseline	6.137	53.67	77.61	81.19	79.15	72.23	45.00	81.01	73.24	70.39
	W2A4KV4	OmniQuant*	3.2e5	25.85	25.46	38.26	25.35	0.00	26.80	51.58	49.09	30.30
		AffineQuant*	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
		SpinQuant-RTN	31.27	24.40	33.50	62.14	36.01	19.99	27.40	56.69	52.80	39.12
		SpinQuant-GPTQ	29.60	22.18	34.89	58.23	35.07	13.42	27.20	55.88	51.22	37.26
		DuQuant	5.4e5	27.30	24.62	50.67	26.49	0.00	28.20	50.82	50.59	32.34
		DuQuant + LWC	4.1e4	25.60	26.09	37.86	25.55	0.00	25.60	51.36	49.25	30.16
		BoA†	**18.23**	28.92	48.86	62.54	44.62	29.38	28.80	62.79	54.22	**45.02**
LLaMA3.2-1B	FP16	Baseline	13.15	38.14	63.26	69.51	60.78	54.38	34.60	74.37	59.51	56.82
	W2A4KV4	OmniQuant*	7.0e3	24.83	26.26	37.98	25.59	0.00	27.60	49.84	50.75	30.36
		AffineQuant*	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
		SpinQuant-RTN	110.6	22.61	32.07	52.14	28.59	4.08	28.20	52.83	49.49	33.75
		SpinQuant-GPTQ	143.8	22.35	31.23	51.96	28.91	2.91	26.20	53.32	51.85	33.59
		DuQuant	5.6e4	26.62	24.71	41.74	26.16	0.01	28.20	48.80	50.51	30.84
		DuQuant + LWC	8.5e3	28.07	26.01	38.26	25.84	0.00	28.80	50.11	50.91	31.00
		BoA†	**77.05**	22.61	36.66	57.22	32.13	8.07	25.80	55.55	50.99	**36.13**
LLaMA3.2-3B	FP16	Baseline	11.04	46.16	67.80	78.62	70.44	62.15	36.00	75.52	67.40	63.01
	W2A4KV4	OmniQuant*	6.6e3	27.82	25.00	38.35	25.43	0.00	28.00	52.18	49.80	30.82
		AffineQuant*	6.0e3	25.94	26.89	38.10	25.66	0.01	28.40	51.58	49.80	30.80
		SpinQuant-RTN	51.19	24.83	31.48	45.32	29.90	11.05	27.80	55.17	50.20	34.47
		SpinQuant-GPTQ	65.09	23.21	33.54	38.26	31.16	7.18	25.80	55.98	51.38	33.31
		DuQuant	1.2e5	26.02	25.51	49.02	26.74	0.00	29.00	51.03	49.88	32.15
		DuQuant + LWC	1.7e3	24.40	26.09	37.83	25.91	0.00	27.80	50.49	52.49	30.63
		BoA†	**37.12**	27.99	37.67	59.30	37.91	15.81	27.40	57.73	52.41	**39.53**

† BoA has been applied after transforming the model via SpinQuant.
* The LET option has been deactivated because this option does not support models exploiting GQA.
** 'NaN' means that loss diverges in the quantization process.
*** Results for other configurations are provided in Appendix D.3 due to the page limitation.

timized with respect to activation-only quantized networks (*i.e.*, weights are fixed with full-precision) as in (Liu et al., 2024).

From Table 7 and Tables 13-15 in Appendix D.3, we observe that the outstanding weight-quantization performance of the proposed BoA leads to the state-of-the-art performance for the weight-activation quantization. In particular, the performance gain is noticeable for low-bit (*e.g.*, W2A4KV4 and W2A4KV16). For example, compared to SpinQuant-GPTQ, BoA achieves 10%p accuracy improvement on LLaMA2-7B and 12.5%p improvement on LLaMA2-13B (see Table 7). Compared to the conventional gradient-based methods (*i.e.* OmniQuant, AffineQuant, and DuQuant), BoA achieves 8%p, 13%p, 13%p, 5%p, and 7%p improvement on LLaMA2-7B, LLaMA2-13B, LLaMA3-8B, LLaMA3.2-1B, and LLaMA3.2-3B, respectively.

5. Conclusion

In this paper, we proposed a novel backpropagation-free PTQ algorithm called BoA. To consider the inter-layer dependencies within the attention module, we approximated the Hessian matrices by exploiting the attention reconstruction error rather than the layer-wise reconstruction error. To mitigate the computational overhead incurred by the proposed attention-aware Hessians, we also incorporated several techniques, including Hessian relaxation, efficient computation of inverse and Cholesky decomposition of attention-aware Hessians, and simultaneous quantization of different attention heads. Finally, from extensive experiments, we demonstrated the efficacy of the proposed BoA.

Acknowledgment

We would like thanks to Daehyun Kim, Ph.D., Hyeonmok Ko, Ph.D., and Hyemi Jang, Ph.D. for their helpful discussion.

Impact Statement

BOA is a highly efficient and accurate quantization algorithm that minimizes accuracy loss while significantly reducing the time required for model quantization and deployment. By synergizing with other methods, BOA facilitates the efficient operation of LLMs on resource-constrained hardware using only integer arithmetic. This breakthrough is particularly impactful for on-device AI, allowing real-time model inference on mobile and edge devices without the need for server access, paving the way for scalable, decentralized AI solutions.

References

Ashkboos, S., Mohtashami, A., Croci, M. L., Li, B., Cameron, P., Jaggi, M., Alistarh, D., Hoefler, T., and Hensman, J. QuaRot: Outlier-free 4-bit inference in rotated LLMs. *arXiv:2404.00456*, 2024.

Bisk, Y., Zellers, R., Gao, J., Choi, Y., et al. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Botev, A., Ritter, H., and Barber, D. Practical Gauss-Newton optimisation for deep learning. In *International Conference on Machine Learning*, pp. 557–565. PMLR, 2017.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. *arXiv:1905.10044*, 2019.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

Dettmers, T., Svirschevski, R., Egiazarian, V., Kuznedelev, D., Frantar, E., Ashkboos, S., Borzunov, A., Hoefler, T., and Alistarh, D. SpQR: A sparse-quantized representation for near-lossless llm weight compression. *arXiv:2306.03078*, 2023.

Edalati, A., Ghaffari, A., Asgharian, M., Hou, L., Chen, B., and Nia, V. P. OAC: Output-adaptive calibration for accurate post-training quantization. *arXiv:2405.15025*, 2024.

Egiazarian, V., Panferov, A., Kuznedelev, D., Frantar, E., Babenko, A., and Alistarh, D. Extreme compression of large language models via additive quantization. *arXiv:2401.06118*, 2024.

Frantar, E. and Alistarh, D. Optimal brain compression: A framework for accurate post-training quantization and pruning. *Advances in Neural Information Processing Systems*, 35:4475–4488, 2022.

Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. OPTQ: Accurate quantization for generative pre-trained Transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

Hubara, I., Nahshan, Y., Hanani, Y., Banner, R., and Soudry, D. Accurate post training quantization with small calibration sets. In *International Conference on Machine Learning*, pp. 4466–4475. PMLR, 2021.

Jeon, Y., Lee, C., Cho, E., and Ro, Y. Mr.BiQ: Post-training non-uniform quantization based on minimizing the reconstruction error. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12329–12338, 2022.

Jeon, Y., Lee, C., and Kim, H.-y. GENIE: show me the data for quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12064–12073, 2023a.

Jeon, Y., Lee, C., Park, K., and Kim, H.-y. A frustratingly easy post-training quantization scheme for LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14446–14461, 2023b.

Jin, J., Liang, C., Wu, T., Zou, L., and Gan, Z. KDLSQ-BERT: A quantized BERT combining knowledge distillation with learned step size quantization. *arXiv preprint arXiv:2101.05938*, 2021.

Kim, J., Lee, C., Cho, E., Park, K., Kim, H.-y., Kim, J., and Jeon, Y. Towards next-level post-training quantization of hyper-scale transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 94292–94326, 2024.

Kim, S., Hooper, C., Gholami, A., Dong, Z., Li, X., Shen, S., Mahoney, M. W., and Keutzer, K. SqueezeLLM: Dense-and-sparse quantization. *arXiv:2306.07629*, 2023.

Li, Y., Gong, R., Tan, X., Yang, Y., Hu, P., Zhang, Q., Yu, F., Wang, W., and Gu, S. BRECQ: Pushing the limit of post-training quantization by block reconstruction. In *International Conference on Learning Representations (ICLR)*, 2021.

Lin, H., Xu, H., Wu, Y., Cui, J., Zhang, Y., Mou, L., Song, L., Sun, Z., and Wei, Y. DuQuant: Distributing outliers via dual transformation makes stronger quantized LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Liu, Z., Oguz, B., Zhao, C., Chang, E., Stock, P., Mehdad, Y., Shi, Y., Krishnamoorthi, R., and Chandra, V. Llm-qat: Data-free quantization aware training for large language models, 2023.

Liu, Z., Zhao, C., Fedorov, I., Soran, B., Choudhary, D., Krishnamoorthi, R., Chandra, V., Tian, Y., and Blankevoort, T. SpinQuant: LLM quantization with learned rotations. *arXiv:2405.16406*, 2024.

Ma, Y., Li, H., Zheng, X., Ling, F., Xiao, X., Wang, R., Wen, S., Chao, F., and Ji, R. AffineQuant: Affine transformation quantization for large language models. *arXiv:2403.12544*, 2024.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *arXiv:1609.07843*, 2016.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv:1809.02789*, 2018.

Nagel, M., Amjad, R. A., Van Baalen, M., Louizos, C., and Blankevoort, T. Up or down? Adaptive rounding for post-training quantization. In *International Conference on Machine Learning (ICML)*, pp. 7197–7206, 2020.

Paperno, D., Kruszewski, G., Lazaridou, A., Pham, Q. N., Bernardi, R., Pezzelle, S., Baroni, M., Boleda, G., and Fernández, R. The LAMBADA dataset: Word prediction requiring a broad discourse context. *arXiv:1606.06031*, 2016.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. WinoGrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Shao, W., Chen, M., Zhang, Z., Xu, P., Zhao, L., Li, Z., Zhang, K., Gao, P., Qiao, Y., and Luo, P. OmniQuant: Omnidirectionally calibrated quantization for large language models. *arXiv:2308.13137*, 2023.

Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., and Liu, Y. RoFormer: Enhanced Transformer with rotary position embedding. *arXiv:2104.09864*, 2023.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. LLaMA: Open and efficient foundation language models. *arXiv:2302.13971*, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*, 2023b.

Tseng, A., Chee, J., Sun, Q., Kuleshov, V., and De Sa, C. QuIP#: Even better LLM quantization with Hadamard incoherence and lattice codebooks. *arXiv:2402.04396*, 2024.

Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., and Han, S. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pp. 38087–38099. PMLR, 2023.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. 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.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. OPT: Open pre-trained Transformer language models. *arXiv:2205.01068*, 2022.

A. Proof of Footnote 2

In our proof, we use the following useful properties of the Kronecker product:

$$\mathrm{vec}\left(\mathbf{M}_1\mathbf{M}_2\mathbf{M}_3\right) = \left(\mathbf{M}_3^T \otimes \mathbf{M}_1\right)\mathrm{vec}(\mathbf{M}_2), \tag{17a}$$

$$\left(\mathbf{M}_1 \otimes \mathbf{M}_2\right)^T = \mathbf{M}_1^T \otimes \mathbf{M}_2^T, \tag{17b}$$

$$\left(\mathbf{M}_1 \otimes \mathbf{M}_2\right)\left(\mathbf{M}_3 \otimes \mathbf{M}_4\right) = \mathbf{M}_1\mathbf{M}_3 \otimes \mathbf{M}_2\mathbf{M}_4, \tag{17c}$$

where $\mathrm{vec}(\cdot)$ denotes the vectorization operation.

Using (17a), we have

$$\|\mathbf{M}_1\Delta\mathbf{W}\mathbf{M}_2\|_F^2 = \left\|\left(\mathbf{M}_2^T \otimes \mathbf{M}_1\right)\Delta\mathbf{w}\right\|_2^2 = \Delta\mathbf{w}^T\left(\mathbf{M}_2^T \otimes \mathbf{M}_1\right)^T\left(\mathbf{M}_2^T \otimes \mathbf{M}_1\right)\Delta\mathbf{w},$$

where $\Delta\mathbf{w} = \mathrm{vec}(\Delta\mathbf{W})$. In addition, by (17b) and (17c), we have

$$\begin{aligned}\Delta\mathbf{w}^T\left(\mathbf{M}_2^T \otimes \mathbf{M}_1\right)^T\left(\mathbf{M}_2^T \otimes \mathbf{M}_1\right)\Delta\mathbf{w} &= \Delta\mathbf{w}^T\left(\mathbf{M}_2 \otimes \mathbf{M}_1^T\right)\left(\mathbf{M}_2^T \otimes \mathbf{M}_1\right)\Delta\mathbf{w} \\ &= \Delta\mathbf{w}^T\left(\mathbf{M}_2\mathbf{M}_2^T \otimes \mathbf{M}_1^T\mathbf{M}_1\right)\Delta\mathbf{w}.\end{aligned}$$

Finally, by exploiting the fact that $\frac{\partial^2\mathbf{x}^T\mathbf{A}\mathbf{x}}{\partial\mathbf{x}^2} = \mathbf{A} + \mathbf{A}^T$, we obtain

$$\begin{aligned}\frac{\partial^2\|\mathbf{M}_1\Delta\mathbf{W}\mathbf{M}_2\|_F^2}{\partial\Delta\mathbf{w}^2} &= \mathbf{M}_2\mathbf{M}_2^T \otimes \mathbf{M}_1^T\mathbf{M}_1 + \left(\mathbf{M}_2\mathbf{M}_2^T \otimes \mathbf{M}_1^T\mathbf{M}_1\right)^T \\ &\overset{(a)}{=} \mathbf{M}_2\mathbf{M}_2^T \otimes \mathbf{M}_1^T\mathbf{M}_1 + \left(\mathbf{M}_2\mathbf{M}_2^T\right)^T \otimes \left(\mathbf{M}_1^T\mathbf{M}_1\right)^T \\ &= 2\mathbf{M}_2\mathbf{M}_2^T \otimes \mathbf{M}_1^T\mathbf{M}_1,\end{aligned}$$

where (a) follows from (17b). This completes the proof.

B. Attention-aware Hessians for Models Exploiting RoPE

As mentioned, when RoPE is applied, the proposed surrogate $\|\mathbf{K}_h\Delta\mathbf{Q}_h^T\|_F^2$ used to develop the attention-aware Hessian $\mathbf{H}^{(\mathbf{w}_{Q,h})}$ in (11) changes to $\|\widetilde{\mathbf{K}}_h\Delta\widetilde{\mathbf{Q}}_h^T\|_F^2$, where $\widetilde{\mathbf{K}}_h = \mathrm{RoPE}(\mathbf{K}_h)$ and $\widetilde{\mathbf{Q}}_h = \mathrm{RoPE}(\mathbf{Q}_h)$. Let $\mathbf{R}_\ell$ be the rotary matrix for the ℓ-th token (see eq. (15) in (Su et al., 2023)) and $\widetilde{\mathbf{Q}}_h^T = [\widetilde{\mathbf{q}}_{h,1}\ \ldots\ \widetilde{\mathbf{q}}_{h,L}]$, then the objective can be expressed as

$$\|\widetilde{\mathbf{K}}_h\Delta\widetilde{\mathbf{Q}}_h^T\|_F^2 = \sum_{\ell=1}^{L}\|\widetilde{\mathbf{K}}_h\Delta\widetilde{\mathbf{q}}_{h,\ell}\|_2^2 = \sum_{\ell=1}^{L}\|\widetilde{\mathbf{K}}_h\Delta(\mathbf{R}_\ell\mathbf{W}_{Q,h}\mathbf{x}_\ell)\|_2^2 = \sum_{\ell=1}^{L}\|\widetilde{\mathbf{K}}_h\mathbf{R}_\ell\Delta\mathbf{W}_{Q,h}\mathbf{x}_\ell\|_2^2,$$

which yields the following Hessian equation (see Footnote 2):

$$\mathbf{H}^{(\mathbf{w}_{Q,h})} = \sum_{\ell=1}^{L}(2\mathbf{x}_\ell\mathbf{x}_\ell^T \otimes \mathbf{R}_\ell^T\widetilde{\mathbf{K}}_h^T\widetilde{\mathbf{K}}_h\mathbf{R}_\ell).$$

Finally, we take the factorized approximation for the summation of Kronecker products (i.e., $\mathbb{E}[\mathbf{M}_1\otimes\mathbf{M}_2] \approx \mathbb{E}[\mathbf{M}_1]\otimes\mathbb{E}[\mathbf{M}_2]$; see eq. (20) in (Botev et al., 2017)):

$$\mathbf{H}^{(\mathbf{w}_{Q,h})} \approx \sum_{\ell=1}^{L}2\mathbf{x}_\ell\mathbf{x}_\ell^T \otimes \frac{1}{L}\sum_{\ell=1}^{L}\mathbf{R}_\ell^T\widetilde{\mathbf{K}}_h^T\widetilde{\mathbf{K}}_h\mathbf{R}_\ell = 2\mathbf{X}\mathbf{X}^T \otimes \frac{1}{L}\sum_{\ell=1}^{L}\mathbf{R}_\ell^T\widetilde{\mathbf{K}}_h^T\widetilde{\mathbf{K}}_h\mathbf{R}_\ell.$$

By taking similar steps, the attention-aware Hessian for the key projection weight $\mathbf{W}_{K,h}$ with RoPE can be established as

$$\mathbf{H}^{(\mathbf{w}_{K,h})} = 2\mathbf{X}\mathbf{X}^T \otimes \frac{1}{L}\sum_{\ell=1}^{L}\mathbf{R}_\ell^T\widetilde{\mathbf{Q}}_h^T\widetilde{\mathbf{Q}}_h\mathbf{R}_\ell.$$

C. Refined Weight-update Formula

We recall that the Hessian-based weight-update formula is given by (Frantar & Alistarh, 2022; Frantar et al., 2023)

$$\boldsymbol{\delta w} = -\frac{w_q - \mathcal{Q}(w_q)}{[\mathbf{U}]_{q,q}}[\mathbf{U}]_{q,:} \text{ where } \mathbf{U} = \text{Chol}(\mathbf{H}^{-1})^T.$$

For the proposed attention-aware Hessians in Table 1, we have

$$\mathbf{U}_h = \mathbf{U}_{\text{col},h} \otimes \mathbf{U}_{\text{row},h},$$

where $\mathbf{U}_{\text{col},h} = \text{Chol}(\mathbf{H}_{\text{col},h}^{-1})^T$ and $\mathbf{U}_{\text{row},h} = \text{Chol}(\mathbf{H}_{\text{row},h}^{-1})^T$ (see Section 3.3). Therefore, the weight-update formula can be recast as

$$\boldsymbol{\delta w}_h = -\frac{w_q - \mathcal{Q}(w_q)}{[\mathbf{U}_{\text{col},h} \otimes \mathbf{U}_{\text{row},h}]_{q,q}}[\mathbf{U}_{\text{col},h} \otimes \mathbf{U}_{\text{row},h}]_{q,:}.$$

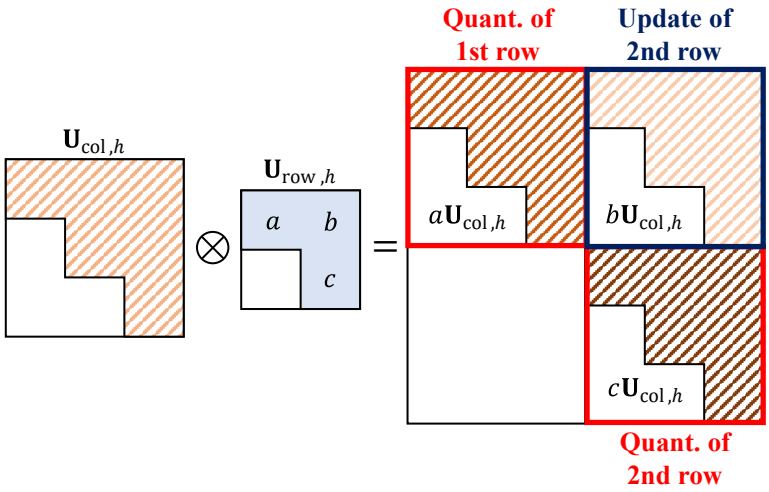


Figure 2. Illustration of the Hessian information when $d_{\text{row}} = 2$ and $d_{\text{col}} = 3$

For simplicity, suppose we quantize the first (0-th) row. When the weight $[\mathbf{W}_h]_{0,j} (= [\mathbf{W}^{(0)}]_{h,j})$ in the j-th column is quantized, the weight-update of the i-th row is simplified as (see Figure 2 for the ease of understanding)

$$\begin{aligned}[\boldsymbol{\delta}\mathbf{W}_h]_{i,:} &= -\frac{[\mathbf{W}_h]_{0,j} - \mathcal{Q}([\mathbf{W}_h]_{0,j})}{[\mathbf{U}_{\text{row},h}]_{0,0}[\mathbf{U}_{\text{col},h}]_{j,j}}[\mathbf{U}_{\text{row},h}]_{0,i}[\mathbf{U}_{\text{col},h}]_{j,:} \\ &= -\frac{[\mathbf{W}_h]_{0,j} - \mathcal{Q}([\mathbf{W}_h]_{0,j})}{[\mathbf{U}_{\text{col},h}]_{j,j}} \cdot \frac{[\mathbf{U}_{\text{row},h}]_{0,i}[\mathbf{U}_{\text{col},h}]_{j,:}}{[\mathbf{U}_{\text{row},h}]_{0,0}}\end{aligned}$$

Thus, after the quantization of all weights in the first row, the total amount of the weight-update for the i-th row can be expressed as

$$\begin{aligned}[\boldsymbol{\delta}\mathbf{W}_{h,\text{total}}]_{i,:} &= -\sum_{j=0}^{d_{\text{col}}-1} \frac{[\mathbf{W}_h]_{0,j} - \mathcal{Q}([\mathbf{W}_h]_{0,j})}{[\mathbf{U}_{\text{col},h}]_{j,j}} \cdot \frac{[\mathbf{U}_{\text{row},h}]_{0,i}[\mathbf{U}_{\text{col},h}]_{j,:}}{[\mathbf{U}_{\text{row},h}]_{0,0}} \\ &= -\frac{[\mathbf{U}_{\text{row},h}]_{0,i}}{[\mathbf{U}_{\text{row},h}]_{0,0}} \sum_{j=0}^{d_{\text{col}}-1} \frac{[\mathbf{W}_h]_{0,j} - \mathcal{Q}([\mathbf{W}_h]_{0,j})}{[\mathbf{U}_{\text{col},h}]_{j,j}} \cdot [\mathbf{U}_{\text{col},h}]_{j,:}.\end{aligned}$$

Furthermore, by noting that (see line 5 in Algorithm 2)

$$[\mathbf{E}_{\text{GPTQ}}]_{h,j} = \frac{[\mathbf{W}_h]_{0,j} - \mathcal{Q}([\mathbf{W}_h]_{0,j})}{[\mathbf{U}_{\text{col},h}]_{j,j}},$$

we obtain

$$[\boldsymbol{\delta}\mathbf{W}_{h,\text{total}}]_{i,:} = -\frac{[\mathbf{U}_{\text{row},h}]_{0,i}}{[\mathbf{U}_{\text{row},h}]_{0,0}} \sum_{j=0}^{d_{\text{col}}-1} [\mathbf{E}_{\text{GPTQ}}]_{h,j} \cdot [\mathbf{U}_{\text{col},h}]_{j,:} = -\frac{[\mathbf{U}_{\text{row},h}]_{0,i}}{[\mathbf{U}_{\text{row},h}]_{0,0}} [\mathbf{E}_{\text{GPTQ}}]_{h,:} \mathbf{U}_{\text{col},h}.$$

As a result, the weight-update matrix to compensate for the quantization error of the first row is given by

$$[\boldsymbol{\delta}\mathbf{W}_{h,\text{total}}]_{0:,:} = -\frac{[\mathbf{U}_{\text{row},h}^T]_{0:,0} [\mathbf{E}_{\text{GPTQ}}]_{h,:} \mathbf{U}_{\text{col},h}}{[\mathbf{U}_{\text{row},h}]_{0,0}}. \tag{18}$$

By taking similar steps as above, we can easily generalize (18) for the j-th row as follows:

$$[\boldsymbol{\delta}\mathbf{W}_{h,\text{total}}]_{j:,:} = -\frac{[\mathbf{U}_{\text{row},h}^T]_{j:,j} [\mathbf{E}_{\text{GPTQ}}]_{h,:} \mathbf{U}_{\text{col},h}}{[\mathbf{U}_{\text{row},h}]_{j,j}}. \tag{19}$$

D. Additional Experimental Results

D.1. Performance of Weight-Rounding Optimization

We evaluate the weight-only quantization performance of the proposed BOA. To solely compare the performance of the weight-rounding optimization, we do not apply any transform (*e.g.*, smoothing, rotation, and permutation) to the models. In this appendix, we provide the results for the INT4 weight quantization of LLaMA2 and LLaMA3 models (see Table 8) and the results on LLaMA1 models (see Table 9) and OPT models (see Table 10).

Table 8. INT4 weight-only quantization performance on LLaMA2 and LLaMA3 models without transformation

Model	Precision	Method	Wiki2 PPL (↓)	Zero-shot Accuracy (↑)								
				Arc-c	Arc-e	BQ	HS	LAMB	OBQA	PIQA	WG	Average
LLaMA2-7B	FP	Baseline	5.473	45.90	74.66	77.92	75.94	70.86	44.00	78.89	68.90	67.13
	INT4	RTN	6.998	41.64	67.05	72.72	71.36	59.78	42.60	77.04	65.75	62.24
		GPTQ	5.809	42.06	69.82	67.06	70.44	62.80	43.00	77.15	68.11	62.56
		BoA	**5.622**	43.94	72.26	78.04	74.53	68.96	44.20	78.40	69.14	**66.18**
LLaMA2-13B	FP	Baseline	4.885	49.06	77.65	80.49	79.38	73.41	45.80	80.69	72.22	69.84
	INT4	RTN	6.319	43.00	66.88	71.01	66.14	62.25	38.20	77.04	63.85	61.05
		GPTQ	5.632	47.18	74.49	74.37	75.00	70.89	44.20	78.40	69.69	66.78
		BoA	**5.096**	47.95	75.80	78.93	77.96	70.61	41.60	79.87	71.35	**68.01**
LLaMA3-8B	FP	Baseline	6.137	53.67	77.61	81.19	79.15	72.23	45.00	81.01	73.24	70.39
	INT4	RTN	7.981	49.06	73.65	73.30	76.62	59.55	45.00	78.29	72.93	66.05
		GPTQ	7.960	49.06	73.44	73.70	69.30	60.19	45.00	79.54	73.56	65.47
		BoA	**6.561**	50.17	77.74	80.31	77.42	70.20	44.40	80.14	73.88	**69.28**
LLaMA3.2-1B	FP	Baseline	13.15	38.14	63.26	69.51	60.78	54.38	34.60	74.37	59.51	56.82
	INT4	RTN	18.81	34.30	57.41	65.20	55.93	32.72	33.00	69.15	56.35	50.51
		GPTQ	16.20	35.41	58.63	64.46	55.69	38.08	32.20	69.04	56.27	51.22
		BoA	**14.29**	36.18	60.82	67.22	59.03	51.65	34.00	71.87	58.09	**54.86**
LLaMA3.2-3B	FP	Baseline	11.04	46.16	67.80	78.62	70.44	62.15	36.00	75.52	67.40	63.01
	INT4	RTN	13.86	43.26	61.24	73.67	68.27	54.18	38.40	71.76	64.01	59.35
		GPTQ	12.41	44.28	63.51	77.00	68.44	57.39	36.60	74.16	67.01	61.05
		BoA	**11.74**	44.11	66.75	77.86	69.42	61.74	36.20	75.08	66.85	**62.25**

Table 9. Weight-only quantization performance on LLaMA models without transformation

Model	Precision	Method	Wiki2 PPL (↓)	Zero-shot Accuracy (↑)								
				Arc-c	Arc-e	BQ	HS	LAMB	OBQA	PIQA	WG	Average
LLaMA-7B	FP	Baseline	5.677	44.71	72.94	75.05	76.21	70.66	44.40	79.16	70.01	66.64
	INT2	RTN	5.9e3	26.37	26.73	39.82	26.37	0.01	26.00	51.80	48.62	30.72
		GPTQ	19.63	25.34	41.58	56.27	37.27	26.24	30.20	61.70	55.56	41.77
		BoA	**10.28**	28.75	52.69	64.19	49.34	41.48	32.60	66.54	59.19	**49.35**
	INT3	RTN	61.31	26.45	50.46	59.76	40.07	14.64	33.00	67.08	55.88	43.42
		GPTQ	18.44	33.53	61.36	64.40	56.22	44.91	35.40	72.09	64.33	54.03
		BoA	**6.267**	39.76	67.68	75.35	71.01	68.92	41.80	77.80	67.48	**63.73**
	INT4	RTN	7.912	41.47	70.29	72.81	73.16	63.81	41.60	77.86	68.27	63.66
		GPTQ	7.791	40.36	66.50	70.58	68.48	61.84	39.00	77.42	68.43	61.58
		BoA	**5.899**	43.77	72.18	75.54	74.75	69.60	44.40	78.62	69.77	**66.08**
LLaMA-13B	FP	Baseline	5.090	47.70	74.75	77.95	79.07	73.66	44.80	80.14	72.77	68.86
	INT2	RTN	2.2e3	25.60	26.01	39.48	26.00	0.00	25.20	49.46	48.54	30.04
		GPTQ	34.63	26.37	43.43	56.67	46.85	33.11	31.40	65.07	57.62	45.07
		BoA	**8.309**	32.76	59.51	69.57	59.52	51.79	35.20	71.22	65.19	**55.60**
	INT3	RTN	134.5	25.00	49.24	56.39	36.91	11.41	29.20	64.20	54.62	40.87
		GPTQ	7.096	42.24	69.74	70.73	73.40	62.99	39.20	77.09	68.11	62.94
		BoA	**5.461**	45.82	71.93	75.96	75.71	70.62	43.40	78.35	70.24	**66.50**
	INT4	RTN	7.742	45.22	69.99	72.42	75.67	65.65	41.60	78.78	70.56	64.99
		GPTQ	6.147	45.82	72.26	74.10	76.24	70.15	43.80	80.20	71.90	66.81
		BoA	**5.189**	47.44	74.62	77.31	78.03	73.19	45.40	80.20	73.09	**68.66**
LLaMA-30B	FP	Baseline	4.101	52.90	78.96	82.69	82.63	75.47	48.20	82.26	75.77	72.36
	INT2	RTN	7.1e3	25.60	26.52	40.67	27.20	0.11	24.60	50.92	49.64	30.66
		GPTQ	9.770	35.41	60.14	64.77	61.54	50.12	38.20	70.35	66.54	55.88
		BoA	**6.669**	37.71	64.73	73.00	64.77	61.28	39.00	73.34	67.40	**60.15**
	INT3	RTN	81.76	30.12	58.21	54.10	42.94	6.71	35.00	69.31	58.25	44.33
		GPTQ	7.064	48.12	71.46	74.89	76.73	66.39	43.20	78.35	70.80	66.24
		BoA	**4.575**	49.57	74.75	81.87	79.21	73.77	46.20	79.38	74.19	**69.87**
	INT4	RTN	5.802	49.40	74.16	78.07	79.48	66.67	44.00	79.87	73.16	68.10
		GPTQ	5.188	51.45	75.67	80.18	79.69	70.38	45.40	79.43	73.64	69.48
		BoA	**4.218**	53.16	78.41	82.72	81.95	75.96	49.40	81.88	75.06	**72.32**

Table 10. Weight-only quantization performance on OPT models without transformation

Model	Precision	Method	Wiki2 PPL (↓)	Zero-shot Accuracy (↑)								
				Arc-c	Arc-e	BQ	HS	LAMB	OBQA	PIQA	WG	Average
OPT-125M	FP	Baseline	27.65	22.78	39.98	55.44	31.35	33.44	28.00	62.02	50.28	40.41
	INT2	RTN	1.0e4	23.89	27.31	37.83	25.96	0.00	27.00	50.49	51.14	30.45
		GPTQ	411.3	22.61	31.52	38.47	27.16	0.18	27.20	52.18	51.70	31.38
		BoA	**85.63**	21.42	32.15	48.20	28.50	6.86	26.60	56.47	49.17	**33.67**
	INT3	RTN	233.9	22.01	33.33	52.94	28.52	2.81	28.40	59.19	51.14	34.79
		GPTQ	50.75	21.67	33.84	58.38	29.41	16.96	25.40	58.98	51.14	36.97
		BoA	**31.95**	22.95	36.83	59.66	30.67	27.10	28.80	60.34	50.83	**39.65**
OPT-350M	FP	Baseline	22.00	23.89	40.36	57.68	36.67	40.47	28.20	64.74	52.33	43.04
	INT2	RTN	4.6e3	24.74	27.65	37.83	26.87	0.00	23.40	52.83	52.09	30.68
		GPTQ	61.93	22.53	33.46	57.28	29.66	10.70	23.80	56.75	51.22	35.68
		BoA	**46.53**	22.01	35.90	62.32	30.43	11.85	26.60	58.98	51.62	**37.46**
	INT3	RTN	35.91	23.38	36.99	52.14	34.00	30.37	27.00	62.02	51.62	39.69
		GPTQ	25.25	22.95	38.47	60.58	35.25	41.38	27.60	62.46	51.22	42.49
		BoA	**23.96**	22.87	38.76	61.22	35.30	38.08	28.40	62.57	53.28	**42.56**
OPT-1.3B	FP	Baseline	14.62	29.52	50.97	57.83	53.70	55.19	33.40	72.47	59.51	51.57
	INT2	RTN	1.6e4	24.23	24.75	51.19	26.51	0.00	29.00	52.29	50.04	32.25
		GPTQ	177.4	22.78	31.78	57.06	28.13	6.37	24.80	53.75	48.70	34.17
		BoA	**31.65**	22.70	37.67	62.17	34.65	18.02	26.80	61.48	51.78	**39.41**
	INT3	RTN	754.8	25.34	34.72	52.17	36.31	10.25	27.20	60.07	51.78	37.23
		GPTQ	19.35	27.13	43.43	54.25	47.23	36.30	29.40	68.01	54.30	45.01
		BoA	**15.77**	27.39	48.74	58.99	50.09	51.50	30.60	69.37	57.30	**49.25**
OPT-2.7B	FP	Baseline	12.47	31.31	54.38	60.40	60.62	59.78	35.20	74.81	61.01	54.69
	INT2	RTN	2.7e5	26.11	26.73	62.17	26.77	0.00	29.20	50.87	51.07	34.12
		GPTQ	135.5	24.66	33.71	60.28	33.62	17.62	27.00	57.73	50.59	38.15
		BoA	**22.30**	25.77	41.62	62.26	42.00	33.76	28.80	62.08	52.96	**43.66**
	INT3	RTN	170.5	25.26	38.34	56.51	36.64	14.61	29.80	66.27	53.04	40.06
		GPTQ	15.64	29.27	49.07	45.90	51.79	42.92	33.00	69.80	57.54	47.41
		BoA	**13.63**	29.10	52.90	65.29	55.86	56.94	34.00	72.03	60.06	**53.27**
OPT-6.7B	FP	Baseline	10.86	34.73	60.02	66.06	67.16	65.50	37.40	76.50	65.43	59.10
	INT2	RTN	2.9e4	25.00	25.38	37.83	26.66	0.00	29.20	50.00	51.46	30.69
		GPTQ	95.95	23.21	33.75	61.77	32.09	16.25	25.80	57.02	52.01	37.74
		BoA	**20.57**	28.24	47.31	61.80	47.00	34.09	33.20	66.87	57.54	**47.01**
	INT3	RTN	53.76	27.73	40.87	50.95	48.67	28.32	31.20	68.17	53.12	43.63
		GPTQ	12.05	32.51	56.40	54.53	62.49	55.99	36.60	74.21	60.85	54.20
		BoA	**11.25**	32.76	57.32	66.85	64.53	63.64	37.60	75.57	64.09	**57.80**
OPT-13B	FP	Baseline	10.13	35.75	61.87	65.84	69.87	65.98	39.00	76.88	65.11	60.04
	INT2	RTN	4.7e4	26.79	26.64	40.40	26.25	0.00	27.40	49.78	50.04	30.91
		GPTQ	49.40	26.28	37.04	62.14	41.60	29.17	27.60	61.04	52.17	42.13
		BoA	**15.34**	29.27	48.95	62.78	53.63	44.71	32.60	68.50	58.64	**49.89**
	INT3	RTN	40.46	27.47	38.72	62.54	51.21	29.44	26.80	63.38	49.88	43.68
		GPTQ	11.11	34.56	59.51	62.45	64.81	58.57	36.00	74.92	63.54	56.80
		BoA	**10.39**	34.30	59.93	69.94	66.95	66.61	37.60	75.95	66.93	**59.78**
OPT-30B	FP	Baseline	9.557	38.05	65.36	70.46	72.27	67.85	40.20	78.07	68.43	62.59
	INT2	RTN	4.2e4	26.79	25.55	37.83	25.88	0.00	28.60	50.60	51.22	30.81
		GPTQ	21.19	27.99	41.96	62.45	46.90	39.48	30.40	65.72	54.22	46.14
		BoA	**11.15**	30.55	55.81	62.81	60.19	60.39	34.80	72.47	62.67	**54.96**
	INT3	RTN	86.63	24.32	35.06	57.98	38.07	12.17	26.00	59.90	52.09	38.20
		GPTQ	10.14	34.90	61.70	66.97	69.34	63.70	40.40	77.26	67.01	60.16
		BoA	**9.744**	36.26	62.63	70.28	70.52	68.86	40.20	77.75	67.17	**61.71**

D.2. Comparison with Transformation-based Methods

We compare the weight-only quantization performances of the proposed BoA and existing transformation-based methods. For conventional methods, we ran the official codes provided by the authors and reported the obtained results. For OmniQuant (Shao et al., 2023) and AffineQuant (Ma et al., 2024), we activated both learnable equivalent transformation (LET) and learnable weight clipping (LWC) options. For AffineQuant, we did not activate the `use-ln-matrix` option because this adds extra affine transformation between the normalization layer and linear layers, which incurs additional processing time during the inference. QuaRot-RTN and QuaRot-GPTQ mean that RTN and GPTQ have been applied for the weight quantization after transforming models via QuaRot (Ashkboos et al., 2024). For DuQuant, we report the results obtained with and without activating the LWC option as in the original paper (Lin et al., 2024). When measuring the performance of the proposed BoA, we also transformed the model for fair comparison. We applied QuaRot for the transformation because QuaRot does not require training and incurs no extra costs during the actual inference (Ashkboos et al., 2024).

In this appendix, we provide the results for the INT3 weight quantization of LLaMA2 and LLaMA3 models (see Table 11) and the results on LLaMA1 models (see Table 12).

Table 11. INT3 weight-only quantization performance on transformed LLaMA2 and LLaMA3 models

Model	Precision	Method	Wiki2 PPL (↓)	Zero-shot Accuracy (↑)								
				Arc-c	Arc-e	BQ	HS	LAMB	OBQA	PIQA	WG	Average
LLaMA2-7B	FP16	Baseline	5.473	45.90	74.66	77.92	75.94	70.86	44.00	78.89	68.90	67.13
	INT3	OmniQuant	6.640	39.59	65.66	69.82	70.09	57.58	38.40	76.01	64.88	60.25
		AffineQuant	6.468	40.36	68.01	70.86	70.32	61.87	38.60	76.12	65.27	61.43
		QuaRot-RTN	129.2	23.46	30.60	37.83	29.30	4.75	24.40	53.10	50.20	31.71
		QuaRot-GPTQ	6.122	40.96	70.16	73.30	71.44	67.63	41.40	77.42	67.48	63.72
		DuQuant	6.831	41.13	67.59	69.36	70.52	62.25	41.40	76.01	65.35	61.70
		DuQuant + LWC	6.226	40.70	69.15	74.77	70.97	64.06	42.40	77.04	66.85	63.24
		BoA†	**5.874**	42.06	71.17	73.73	72.29	69.85	41.60	77.37	67.48	**64.44**
LLaMA2-13B	FP16	Baseline	4.885	49.06	77.65	80.49	79.38	73.41	45.80	80.69	72.22	69.84
	INT3	OmniQuant	5.593	44.80	71.72	75.29	75.63	64.94	43.20	78.67	69.30	65.44
		AffineQuant	5.526	45.73	73.74	77.34	74.79	65.56	43.00	77.15	67.01	65.54
		QuaRot-RTN	48.06	22.01	34.72	62.14	33.44	7.80	26.60	57.67	50.51	36.86
		QuaRot-GPTQ	5.382	47.01	75.42	78.84	75.70	72.14	44.60	78.56	70.01	67.79
		DuQuant	5.749	44.88	72.10	79.91	75.37	68.23	43.00	77.80	70.72	66.50
		DuQuant + LWC	5.414	46.67	74.62	77.92	75.93	68.63	43.80	79.98	68.75	67.04
		BoA†	**5.202**	48.29	76.39	79.45	76.23	73.19	45.20	78.67	70.96	**68.55**
LLaMA3-8B	FP16	Baseline	6.137	53.67	77.61	81.19	79.15	72.23	45.00	81.01	73.24	70.39
	INT3	OmniQuant	11.75	37.37	60.94	65.57	66.81	33.02	34.80	71.49	59.59	53.70
		AffineQuant	11.63	37.97	63.64	64.92	67.42	33.95	36.20	71.98	60.30	54.55
		QuaRot-RTN	38.64	23.72	38.22	51.28	47.53	23.63	33.60	62.35	62.04	42.80
		QuaRot-GPTQ	7.490	47.87	74.49	78.44	74.67	67.43	40.00	78.07	72.38	66.67
		DuQuant	11.35	35.84	53.37	64.56	65.56	49.52	36.20	72.25	65.90	55.40
		DuQuant + LWC	10.78	31.66	46.59	64.65	65.25	41.29	38.00	70.13	61.80	52.42
		BoA†	**7.145**	49.06	77.40	80.49	74.54	69.20	43.00	78.51	72.53	**68.09**
LLaMA3.2-1B	FP16	Baseline	13.15	38.14	63.26	69.51	60.78	54.38	34.60	74.37	59.51	56.82
	INT3	OmniQuant	26.61	31.31	51.39	63.67	49.18	18.83	29.60	67.57	55.49	45.88
		AffineQuant	26.43	32.00	52.48	64.31	48.71	20.67	26.20	67.57	52.88	45.60
		QuaRot-RTN	98.24	24.06	36.03	61.96	35.31	7.68	28.60	59.14	51.62	38.05
		QuaRot-GPTQ	16.56	32.42	57.95	64.10	52.48	43.79	32.20	68.99	57.14	51.13
		DuQuant	2.1e4	25.26	25.67	41.35	26.68	0.00	27.80	49.51	48.78	30.63
		DuQuant + LWC	2.7e4	25.68	26.73	40.73	25.34	0.00	29.00	49.40	52.09	31.12
		BoA†	**15.73**	32.42	57.95	66.24	53.86	48.74	33.20	70.18	57.06	**52.46**
LLaMA3.2-3B	FP16	Baseline	11.04	46.16	67.80	78.62	70.44	62.15	36.00	75.52	67.40	63.01
	INT3	OmniQuant	16.97	36.01	57.58	71.47	59.57	39.17	34.00	69.31	60.22	53.42
		AffineQuant	16.79	36.43	56.57	72.97	59.19	40.07	35.20	70.13	60.06	53.83
		QuaRot-RTN	89.54	22.01	32.79	59.08	36.18	4.82	25.00	54.35	53.67	35.99
		QuaRot-GPTQ	13.58	38.31	58.96	75.72	64.59	54.59	35.80	70.18	64.96	57.89
		DuQuant	18.92	30.55	48.32	71.71	60.34	40.54	32.40	66.54	60.38	51.35
		DuQuant + LWC	15.18	37.20	58.33	72.02	61.22	44.58	33.40	70.02	59.98	54.59
		BoA†	**12.97**	42.49	65.74	78.90	65.59	58.40	34.80	73.07	63.46	**60.31**

† BoA has been applied after transforming the model via QuaRot.
* The LET option has been deactivated because this option does not support models exploiting GQA.

Table 12. Weight-only quantization performance on transformed LLaMA models

Model	Precision	Method	Wiki2 PPL (↓)	Zero-shot Accuracy (↑)								
				Arc-c	Arc-e	BQ	HS	LAMB	OBQA	PIQA	WG	Average
LLaMA-7B	FP	Baseline	5.677	44.71	72.94	75.05	76.21	70.66	44.40	79.16	70.01	66.64
	INT2	OmniQuant	15.74	26.28	45.71	61.93	42.66	17.43	29.00	62.40	52.17	42.20
		AffineQuant	11.93	27.22	49.33	62.51	49.26	23.57	32.20	62.89	54.06	45.13
		QuaRot-RTN	1.0e4	27.90	26.39	37.83	26.03	0.00	21.60	49.24	49.49	29.81
		QuaRot-GPTQ	11.53	26.45	45.75	62.48	48.23	37.37	32.80	65.67	57.46	47.03
		DuQuant	9.4e3	28.33	26.89	46.42	26.24	0.00	21.20	50.71	51.62	31.43
		DuQuant + LWC	12.39	27.73	49.37	63.03	49.88	23.65	31.80	64.96	56.75	45.90
		BoA†	**9.812**	30.63	54.88	63.55	52.20	46.50	33.00	68.72	61.01	**51.31**
	INT3	OmniQuant	6.463	38.74	66.96	72.35	69.84	62.79	40.00	77.20	66.69	61.82
		AffineQuant	6.392	39.68	66.46	72.87	70.52	64.99	39.40	77.42	65.04	62.05
		QuaRot-RTN	24.03	29.86	53.11	61.13	50.30	24.53	31.40	69.37	57.30	47.13
		QuaRot-GPTQ	6.166	43.09	70.37	72.02	72.18	65.34	43.20	78.29	68.19	64.09
		DuQuant	6.976	37.37	64.23	71.13	70.81	60.45	39.20	76.39	65.98	60.70
		DuQuant + LWC	6.328	38.74	66.20	73.73	71.24	63.78	42.40	77.31	68.51	62.74
		BoA†	**6.091**	41.04	68.18	71.77	72.26	69.32	40.40	78.18	68.67	**63.73**
LLaMA-13B	FP	Baseline	5.090	47.70	74.75	77.95	79.07	73.66	44.80	80.14	72.77	68.86
	INT2	OmniQuant	13.29	29.69	53.79	62.29	55.31	19.52	31.60	70.29	57.14	47.45
		AffineQuant	10.65	28.75	53.54	65.57	53.03	30.73	34.00	68.12	59.35	49.14
		QuaRot-RTN	4.1e3	28.16	26.47	37.83	25.72	0.00	22.60	51.25	48.46	30.06
		QuaRot-GPTQ	9.331	31.06	58.59	63.79	54.96	45.54	35.20	69.97	63.30	52.80
		DuQuant	6.2e3	25.94	25.93	39.66	26.70	0.00	23.60	50.87	51.62	30.54
		DuQuant + LWC	8.770	31.74	58.38	66.27	60.98	44.46	37.00	72.20	62.19	54.15
		BoA†	**8.341**	33.19	63.64	71.44	60.75	56.30	37.00	71.93	65.90	**57.52**
	INT3	OmniQuant	5.671	45.05	71.13	75.44	75.35	66.82	44.40	78.56	69.46	65.78
		AffineQuant	5.615	43.00	70.79	74.83	74.99	68.95	43.40	79.76	69.38	65.64
		QuaRot-RTN	7.082	37.37	60.14	73.18	67.93	56.42	35.60	76.22	66.14	59.13
		QuaRot-GPTQ	5.471	44.37	71.63	76.94	75.59	72.45	43.00	78.02	71.51	66.69
		DuQuant	5.923	43.94	70.12	74.40	75.76	68.33	42.00	78.56	72.06	65.65
		DuQuant + LWC	5.554	45.31	71.97	72.78	75.85	69.81	45.00	79.49	70.64	66.36
		BoA†	**5.411**	45.56	72.81	75.60	75.95	72.71	45.20	79.27	71.03	**67.27**
LLaMA-30B	FP	Baseline	4.101	52.90	78.96	82.69	82.63	75.47	48.20	82.26	75.77	72.36
	INT2	OmniQuant	8.598	33.45	57.37	62.84	59.21	39.36	37.20	70.02	60.14	52.45
		AffineQuant	7.267	38.14	65.07	73.33	67.61	53.75	38.60	74.21	64.96	59.46
		QuaRot-RTN	3.8e3	25.77	26.52	37.83	25.80	0.01	23.80	51.69	47.83	29.91
		QuaRot-GPTQ	7.283	39.08	68.01	65.44	65.54	58.15	38.40	73.39	67.56	59.45
		DuQuant	5.7e3	27.13	26.98	38.75	26.76	0.00	26.20	49.89	49.17	30.61
		DuQuant + LWC	7.706	38.48	64.60	70.00	67.56	48.08	36.40	72.96	65.11	57.90
		BoA†	**6.525**	41.47	68.01	64.65	67.63	66.31	41.00	74.21	70.32	**61.70**
	INT3	OmniQuant	4.766	50.09	76.18	79.91	79.58	70.79	45.20	79.92	74.43	69.51
		AffineQuant	4.729	50.60	76.81	79.85	79.60	72.83	44.60	80.52	74.74	69.94
		QuaRot-RTN	6.355	37.80	65.32	67.98	64.79	47.46	38.80	72.80	67.17	57.77
		QuaRot-GPTQ	4.759	49.23	77.19	81.22	79.97	75.98	45.20	80.41	75.61	70.60
		DuQuant	5.066	49.49	74.71	79.97	79.87	72.78	46.40	79.54	74.51	69.66
		DuQuant + LWC	4.634	49.32	75.17	81.35	80.26	72.29	46.20	81.01	73.72	69.92
		BoA†	**4.602**	52.73	77.23	80.83	80.04	75.91	45.60	80.79	76.16	**71.16**

† BoA has been applied after transforming the model via QuaRot.

D.3. Weight-Activation Quantization Results

We evaluate the weight-activation quantization performances of the proposed BoA. As in prior works (Shao et al., 2023; Ma et al., 2024; Lin et al., 2024; Ashkboos et al., 2024; Liu et al., 2024), we apply per-token nearest-rounding quantization for input activations and KV caches. We use the notation 'WxAyKVz' to denote the x-bit weight quantization, y-bit input activation quantization, and z-bit KV cache quantization. When measuring the performance of RTN, GPTQ, and the proposed BoA, we use SpinQuant for the model transformation as SpinQuant outperforms QuaRot by optimizing the transformation for the activation quantization (Liu et al., 2024). When optimizing the rotation matrix in SpinQuant, we quantize both weights and activations for SpinQuant-RTN and quantize only activations for SpinQuant-GPTQ and the proposed BoA, as in the original paper (Liu et al., 2024). In this appendix, we provide the results on LLaMA3 models for various quantization configurations.

Table 13. Weight-activation quantization performance on the transformed LLaMA3-8B

Precision	Method	Wiki2 PPL (↓)	Zero-shot Accuracy (↑)								
			Arc-c	Arc-e	BQ	HS	LAMB	OBQA	PIQA	WG	Average
FP16	Baseline	6.137	53.67	77.61	81.19	79.15	72.23	45.00	81.01	73.24	70.39
W2A4KV16	OmniQuant	2.9e5	25.94	26.22	38.50	24.86	0.00	28.00	50.44	49.33	30.41
	AffineQuant	3.2e5	26.62	26.30	38.13	25.62	0.00	29.60	49.51	52.57	31.04
	SpinQuant-RTN	28.38	25.77	35.52	62.11	37.55	18.89	24.20	56.42	51.85	39.04
	SpinQuant-GPTQ	26.35	23.38	38.30	60.43	37.34	16.44	26.20	58.71	53.51	39.29
	DuQuant	4.5e5	25.77	25.97	43.09	26.09	0.00	27.00	51.85	50.83	31.33
	DuQuant + LWC	4.4e4	26.28	27.10	38.38	25.81	0.00	28.00	50.49	50.12	30.77
	BoA†	**17.31**	27.82	47.52	63.27	44.54	27.27	29.40	61.86	54.54	**44.53**
W3A3KV16	OmniQuant	2.7e4	26.79	26.05	38.62	25.43	0.00	26.20	50.16	50.36	30.45
	AffineQuant	3.0e4	26.45	25.25	38.62	25.20	0.00	30.00	50.76	49.41	30.71
	SpinQuant-RTN	21.27	27.05	42.93	62.75	45.02	27.07	29.00	62.02	52.25	43.51
	SpinQuant-GPTQ	20.49	26.96	40.40	54.77	50.47	26.17	31.80	62.13	53.35	43.26
	DuQuant	280.8	23.29	29.42	41.41	30.84	1.37	26.40	51.09	51.22	31.88
	DuQuant + LWC	783.1	22.35	27.57	38.07	28.05	0.18	27.00	51.74	50.59	30.69
	BoA†	**17.56**	31.66	45.88	57.98	53.55	32.56	32.00	65.34	52.17	**46.39**
W4A4KV16	OmniQuant	142.4	22.44	35.02	41.65	33.69	2.46	27.60	54.90	51.14	33.61
	AffineQuant	144.5	25.43	32.70	45.17	33.72	2.83	25.20	56.58	49.96	33.95
	SpinQuant-RTN	8.229	45.14	72.10	75.44	74.07	59.53	40.80	76.44	67.88	63.93
	SpinQuant-GPTQ	7.636	46.33	72.31	75.17	72.69	64.27	42.80	76.44	68.27	64.79
	DuQuant	7.793	45.90	71.46	72.60	74.36	66.06	42.40	78.02	69.61	65.05
	DuQuant + LWC	8.066	44.20	71.13	74.01	73.59	58.55	40.60	76.33	66.77	63.15
	BoA†	**7.496**	47.10	72.05	74.98	74.22	66.29	42.00	76.33	69.53	**65.31**
W4A4KV4	OmniQuant	188.2	22.78	32.32	43.39	31.69	2.01	25.40	54.79	50.20	32.82
	AffineQuant	221.5	22.61	31.23	40.98	31.06	1.53	24.60	55.77	50.28	32.26
	SpinQuant-RTN	8.503	42.06	69.40	72.78	72.69	60.50	38.40	74.21	65.19	61.90
	SpinQuant-GPTQ	7.869	45.56	73.32	71.07	73.88	63.13	39.80	77.04	68.59	64.05
	DuQuant	8.000	45.99	70.88	74.01	73.67	64.32	41.00	76.55	68.90	64.42
	DuQuant + LWC	8.402	44.80	70.29	73.82	73.69	58.61	39.00	75.08	66.77	62.76
	BoA†	**7.705**	45.22	74.54	72.81	74.09	65.72	43.60	77.53	66.69	**65.03**

† BoA has been applied after transforming the model via SpinQuant.
* The LET option has been deactivated for OmniQuant and AffineQuant because this option does not support models exploiting GQA.

Table 14. Weight-activation quantization performance on the transformed LLaMA3.2-1B

Precision	Method	Wiki2 PPL (↓)	Zero-shot Accuracy (↑)								
			Arc-c	Arc-e	BQ	HS	LAMB	OBQA	PIQA	WG	Average
FP16	Baseline	13.15	38.14	63.26	69.51	60.78	54.38	34.60	74.37	59.51	56.82
W2A4KV16	OmniQuant	2.5e3	24.83	26.85	37.83	26.01	0.00	29.00	52.34	51.78	31.08
	AffineQuant	6.1e3	25.00	26.73	37.89	25.64	0.00	26.80	52.18	50.36	30.58
	SpinQuant-RTN	93.90	23.63	32.62	54.59	28.71	7.51	24.60	53.43	50.75	34.48
	SpinQuant-GPTQ	104.4	21.59	32.28	50.83	30.11	7.55	26.20	53.75	49.96	34.03
	DuQuant	1.0e5	26.28	25.04	46.61	26.39	0.00	29.40	49.13	47.51	31.30
	DuQuant + LWC	1.0e4	26.88	24.54	38.04	25.78	0.00	28.00	50.98	50.43	30.58
	BoA†	**59.95**	23.98	37.04	54.68	32.02	9.33	27.00	56.04	52.41	**36.56**
W3A3KV16	OmniQuant	7.1e3	26.79	26.60	38.20	25.24	0.00	29.20	51.03	48.93	30.75
	AffineQuant	5.6e3	27.39	26.77	38.38	25.59	0.00	29.40	49.89	49.09	30.81
	SpinQuant-RTN	57.43	24.23	33.75	53.76	32.59	12.11	25.40	53.86	49.72	35.68
	SpinQuant-GPTQ	57.52	23.38	36.07	48.93	35.13	10.89	27.40	55.82	47.12	35.59
	DuQuant	2.9e4	25.09	26.14	40.00	25.94	0.01	28.40	50.49	50.75	30.85
	DuQuant + LWC	1.2e4	25.17	25.21	39.33	25.65	0.00	27.20	51.85	51.22	30.70
	BoA†	**48.47**	23.81	38.97	53.21	35.55	13.90	26.80	54.84	52.49	**37.45**
W4A4KV16	OmniQuant	156.0	24.23	32.66	47.89	31.15	0.84	27.20	54.84	50.12	33.62
	AffineQuant	155.8	23.63	35.14	47.86	30.61	0.97	28.80	54.52	48.30	33.73
	SpinQuant-RTN	17.66	32.25	54.80	63.30	53.38	40.69	30.80	68.93	53.75	49.74
	SpinQuant-GPTQ	16.68	33.79	55.56	64.77	55.08	41.35	33.40	68.28	54.85	50.89
	DuQuant	2.3e4	26.37	26.47	40.28	26.40	0.01	30.00	48.59	48.38	30.81
	DuQuant + LWC	1.9e4	26.19	26.81	40.31	25.48	0.00	26.00	50.76	47.12	30.33
	BoA†	**16.25**	33.62	58.12	65.72	55.06	44.28	31.80	69.80	55.64	**51.76**
W4A4KV4	OmniQuant	219.2	23.55	31.82	45.47	29.46	0.79	28.00	51.52	48.22	32.35
	AffineQuant	222.0	23.46	32.74	48.53	29.17	0.64	30.20	52.88	52.64	33.78
	SpinQuant-RTN	19.68	32.25	52.86	61.71	50.74	34.91	31.60	66.65	53.43	48.02
	SpinQuant-GPTQ	18.31	30.72	55.18	61.90	52.91	39.13	31.80	67.30	51.93	48.86
	DuQuant	2.0e4	25.60	25.97	39.91	26.03	0.00	27.80	49.02	50.36	30.59
	DuQuant + LWC	1.7e4	27.05	26.01	38.96	26.07	0.01	28.40	49.73	49.41	30.71
	BoA†	**17.83**	32.94	56.02	64.16	53.16	40.41	31.60	68.93	56.04	**50.41**

† BoA has been applied after transforming the model via SpinQuant.
* The LET option has been deactivated for OmniQuant and AffineQuant because this option does not support models exploiting GQA.

Table 15. Weight-activation quantization performance on the transformed LLaMA3.2-3B

Precision	Method	Wiki2 PPL (↓)	Zero-shot Accuracy (↑)								
			Arc-c	Arc-e	BQ	HS	LAMB	OBQA	PIQA	WG	Average
FP16	Baseline	11.04	46.16	67.80	78.62	70.44	62.15	36.00	75.52	67.40	63.01
W2A4KV16	OmniQuant	5.8e3	25.94	26.81	37.89	25.83	0.00	28.80	50.44	49.80	30.69
	AffineQuant	5.6e3	23.89	26.64	38.41	25.74	0.00	26.00	49.56	49.49	29.97
	SpinQuant-RTN	46.31	24.06	31.14	51.87	31.41	12.70	26.60	55.39	50.99	35.52
	SpinQuant-GPTQ	68.74	23.12	32.45	38.29	31.84	9.53	27.20	54.95	51.30	33.59
	DuQuant	1.2e5	25.94	24.75	39.91	26.77	0.00	29.20	51.63	49.80	31.00
	DuQuant + LWC	1.2e3	24.91	25.63	37.83	26.16	0.00	28.40	52.72	48.07	30.47
	BoA†	**34.25**	24.57	36.28	60.24	38.57	17.38	29.40	57.56	52.49	**39.56**
W3A3KV16	OmniQuant	1.2e4	25.00	26.56	37.95	25.84	0.00	27.20	49.73	51.14	30.43
	AffineQuant	1.0e4	25.77	25.72	38.13	26.21	0.00	28.20	50.05	49.72	30.48
	SpinQuant-RTN	36.37	24.23	35.52	55.66	36.04	15.19	26.80	57.78	49.72	37.62
	SpinQuant-GPTQ	27.23	28.33	40.70	60.73	44.15	21.53	28.20	58.16	54.06	41.98
	DuQuant	564.6	22.01	27.02	43.21	29.68	0.94	27.00	50.60	52.80	31.66
	DuQuant + LWC	82.65	23.04	31.61	48.75	33.24	3.01	28.80	53.32	53.43	34.40
	BoA†	**23.76**	28.75	44.53	60.98	46.08	29.32	28.40	60.07	54.22	**44.04**
W4A4KV16	OmniQuant	131.8	24.57	34.55	48.75	36.10	3.21	27.00	56.37	51.07	35.20
	AffineQuant	131.8	23.55	34.60	47.65	35.80	3.01	27.80	56.26	49.57	34.78
	SpinQuant-RTN	12.42	38.23	60.94	72.69	64.68	55.02	31.60	71.22	61.09	56.93
	SpinQuant-GPTQ	11.87	39.51	63.05	74.31	66.42	56.57	35.80	71.22	62.83	58.71
	DuQuant	13.91	38.40	59.93	74.59	66.43	53.84	36.60	70.73	60.46	57.62
	DuQuant + LWC	13.32	38.23	60.73	75.29	65.93	51.58	36.40	71.87	63.38	57.93
	BoA†	**11.57**	40.70	63.68	75.50	66.86	56.77	36.00	70.24	63.61	**59.17**
W4A4KV4	OmniQuant	169.1	23.89	34.22	43.76	32.36	1.62	26.20	54.24	50.51	33.35
	AffineQuant	197.1	21.42	32.87	42.20	31.70	1.25	27.00	55.22	50.75	32.80
	SpinQuant-RTN	12.96	39.16	61.87	72.42	63.06	48.84	34.40	69.70	58.80	56.03
	SpinQuant-GPTQ	12.24	39.33	61.91	72.32	65.56	54.57	35.20	70.13	61.33	57.54
	DuQuant	14.65	39.51	60.02	72.08	65.76	52.54	34.00	69.53	62.19	56.95
	DuQuant + LWC	13.84	38.99	59.68	72.05	64.76	49.18	35.60	70.40	61.56	56.53
	BoA†	**11.98**	39.08	63.64	74.22	65.60	55.51	38.80	71.22	63.14	**58.90**

† BoA has been applied after transforming the model via SpinQuant.
* The LET option has been deactivated for OmniQuant and AffineQuant because this option does not support models exploiting GQA.

E. Pseudocode for GPTQ

In this appendix, we provide the pseudocode of the conventional GPTQ (Frantar et al., 2023), which is omitted in the main manuscript due to the page limitation.

Algorithm 2 GPTQ

Input: weights $\mathbf{W}$, Hessian information $\mathbf{U}_{\text{col}}$, and pre-determined step size $\mathbf{S}$

Initialize quantized output: $\mathbf{Q} \leftarrow \mathbf{0}_{d_{\text{row}} \times d_{\text{col}}}$
Initialize quantization errors: $\mathbf{E} \leftarrow \mathbf{0}_{d_{\text{row}} \times d_{\text{col}}}$
for $j = 0, \cdots, d_{\text{col}} - 1$ **do**
    Quantize the j-th column: $\mathbf{Q}_{:,j} \leftarrow \text{quant}(\mathbf{W}_{:,j}, \mathbf{S})$
    Estimate quantization error: $\mathbf{E}_{:,j} \leftarrow (\mathbf{W}_{:,j} - \mathbf{Q}_{:,j}) / [\mathbf{U}_{\text{col}}]_{j,j}$
    Update weights: $\mathbf{W}_{:,j:} \leftarrow \mathbf{W}_{:,j:} - \mathbf{E}_{:,j} \cdot [\mathbf{U}_{\text{col}}]_{j,j:}$
end for

Output: quantized weights $\mathbf{Q}$, quantization error $\mathbf{E}$