GuidedQuant: Large Language Model Quantization via Exploiting End-Loss Guidance

Jinuk Kim[1,2] Marwa El Halabi[3] Wonyo Park[4] Clemens JS Schaefer[4]
Deokjae Lee[1,2] Yeonhong Park[1] Jae W. Lee[1] Hyun Oh Song[1,2]

[1]Seoul National University [2]Neural Processing Research Center
[3]Samsung AI Lab, Montreal [4]Google

ICML 2025

Demo: GuidedQuant

Llama-3.3-70B-Instruct (2-bit quantized)

Input: Write me a short and concise story about Harry, Ron, and Hermione.

SqueezeLLM

Harry, Ron, and Hermione were the best of friends.
They had been friends since they were kids, and their friendship had been strong since then.
They had been through a lot together, and their bond had grown even stronger.
They had been through a lot, and
they had been through a lot of things, and
they had been through a lot of things.
They had been through a lot, and
they had been through a lot of things, and
they had been through a lot of things.
They had been through a lot, and
they had been through a lot of things, and

GuidedQuant + LNQ (Ours)

As the sun set over the Hogwarts grounds, Harry, Ron, and Hermione sat together on the grass, watching the stars twinkle to life.
They had just finished a long day of classes, and were now unwinding.
Ron began to tell them about his latest Quidditch match, and they all shared a laugh.
The three of them had been friends since their first year and had always been there to support each other.
Now, as they sat together, they each knew that they would always be there for one another.

Problem: Post-training Quantization for LLMs

There are two dominant PTQ approaches for LLM quantization:

- **Layer-wise Output-based Quantization:** Minimize the mean-squared error between each layer's original and quantized outputs.

$$\|\mathbf{X}^{(l)}\mathbf{W}^{(l)} - \mathbf{X}^{(l)}\widehat{\mathbf{W}}^{(l)}\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{d_{\text{out}}^{(l)}}\left(Z_{ij}^{(l)} - \widehat{Z}_{ij}^{(l)}\right)^2$$

Problem: Post-training Quantization for LLMs

There are two dominant PTQ approaches for LLM quantization:

- **Layer-wise Output-based Quantization:** Minimize the mean-squared error between each layer's original and quantized outputs.

$$\|\mathbf{X}^{(l)}\mathbf{W}^{(l)} - \mathbf{X}^{(l)}\widehat{\mathbf{W}}^{(l)}\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{d_{\text{out}}^{(l)}} \left(Z_{ij}^{(l)} - \widehat{Z}_{ij}^{(l)}\right)^2$$

- **Diagonal Fisher Information Matrix (FIM):** Weigh the individual weight errors by gradients of the end loss.

$$(\widehat{\mathbf{w}} - \mathbf{w})^\top \mathrm{diag}(\mathbf{F})(\widehat{\mathbf{w}} - \mathbf{w}) = \sum_k \left(\frac{\partial \ell}{\partial w_k}\right)^2 (\widehat{w}_k - w_k)^2$$

Problem: Post-training Quantization for LLMs

- **Layer-wise Output-based Quantization:** Minimize the mean-squared error between each layer's original and quantized outputs.

$$\|\mathbf{X}^{(l)}\mathbf{W}^{(l)} - \mathbf{X}^{(l)}\widehat{\mathbf{W}}^{(l)}\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{d_{\text{out}}^{(l)}}\left(Z_{ij}^{(l)} - \widehat{Z}_{ij}^{(l)}\right)^2$$

$\rightarrow$ Treats all hidden features equally, ignoring how much each feature contributes to the end loss.

Problem: Post-training Quantization for LLMs

- **Diagonal Fisher Information Matrix (FIM):** Weigh the individual weight errors by gradients of the end loss.

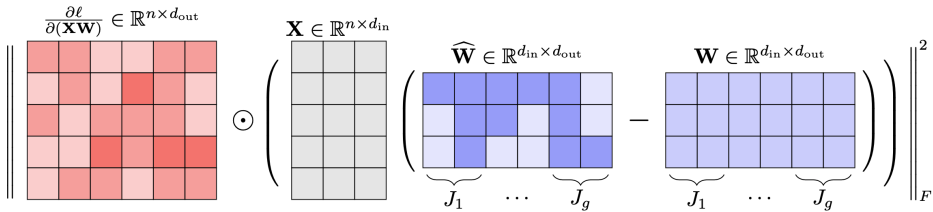
$$\begin{aligned}
\ell(\widehat{\mathbf{w}}) - \ell(\mathbf{w}) &\approx (\widehat{\mathbf{w}} - \mathbf{w})^\top \mathbf{F} (\widehat{\mathbf{w}} - \mathbf{w}) && \text{(2nd order Taylor expansion)} \\
&\approx (\widehat{\mathbf{w}} - \mathbf{w})^\top \mathrm{diag}(\mathbf{F}) (\widehat{\mathbf{w}} - \mathbf{w}) && \text{(Diagonal Approximation)} \\
&= \sum_k \left(\frac{\partial \ell}{\partial w_k} \right)^2 (\widehat{w}_k - w_k)^2
\end{aligned}$$

$\rightarrow$ Approximates FIM with only its diagonal, discarding cross-weight interactions, which are crucial.

GuidedQuant: Objective

GuidedQuant bridges these gaps by **integrating end loss gradients** into the quantization objective while **preserving intra-channel dependencies**.

$$\left\| \frac{\partial \ell}{\partial \mathbf{Z}^{(l)}} \odot (\mathbf{X}^{(l)}\mathbf{W}^{(l)} - \mathbf{X}^{(l)}\widehat{\mathbf{W}}^{(l)}) \right\|_F^2 = n \sum_{l=1}^{L} \sum_{j=1}^{d_{\text{out}}^{(l)}} (\mathbf{w}_j^{(l)} - \widehat{\mathbf{w}}_j^{(l)})^\top \mathbf{F}_j^{(l)} (\mathbf{w}_j^{(l)} - \widehat{\mathbf{w}}_j^{(l)}).$$



GuidedQuant: Objective

GuidedQuant bridges these gaps by **integrating end loss gradients** into the quantization objective while **preserving intra-channel dependencies**.

$$\left\| \frac{\partial \ell}{\partial \mathbf{Z}^{(l)}} \odot (\mathbf{X}^{(l)}\mathbf{W}^{(l)} - \mathbf{X}^{(l)}\widehat{\mathbf{W}}^{(l)}) \right\|_F^2 = n \sum_{l=1}^{L} \sum_{j=1}^{d_{\text{out}}^{(l)}} (\mathbf{w}_j^{(l)} - \widehat{\mathbf{w}}_j^{(l)})^\top \mathbf{F}_j^{(l)} (\mathbf{w}_j^{(l)} - \widehat{\mathbf{w}}_j^{(l)}).$$

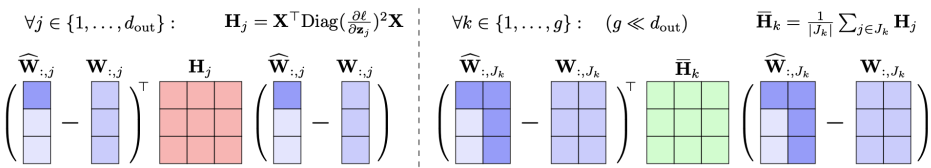
This objective corresponds to **block-diagonal FIM approximation**, and is a more accurate approximation than

→ **Layer-wise output error** objective which assumes $\frac{\partial \ell}{\partial \mathbf{Z}^{(l)}} \propto \mathbf{I}$.
→ **Diagonal FIM** objective which ignores off-diagonal entries.

GuidedQuant: Averaging Approximation

- However, $\mathbf{H}_j^{(l)} := \mathbf{F}_j^{(l)}$ *depends* on each output channel $j \in \{1, \ldots, d_{\text{out}}^{(l)}\}$, making it infeasible to compute and store.
- Solution: Group d_{out} channels into $g \ll d_{\text{out}}$ clusters $(J_1, \ldots, J_g)$, and average Fisher blocks within each group.

$$\overline{\mathbf{H}}_k^{(l)} = \frac{1}{|J_k|} \sum_{j \in J_k} \mathbf{H}_j^{(l)}.$$



GuidedQuant: Results

GuidedQuant can be plugged into any layer-wise PTQ backend.

Type	Method	Bits↓	Wiki2-4K↓
	Original	16	5.12
Weight-only Scalar	SqueezeLLM	2.01	39.58
	LNQ (Ours)	2.01	23.31
	LNQ + GQuant (Ours)	2.01	**8.83**
Weight-only Vector	QTIP	2.00	6.82
	QTIP + GQuant (Ours)	2.00	**6.11**
Type	Method	Bits↓	Wiki2-2K↓
	Original	16	5.47
Weight-and-Activation	SpinQuant	W4A4KV4	5.95
	SpinQuant + GQuant (Ours)	W4A4KV4	**5.89**

→ Improves state-of-the-art methods for **weight-only scalar**, **weight-only vector**, and **weight-and-activation quantization**.

LNQ: Problem Formulation

Regarding **weight-only scalar quantization**, we further propose non-uniform quantization method **LNQ (Layer-wise Non-uniform Quantization)**.

The optimization problem involves discrete assignment $\mathbf{P}^{(j)} \in \{0,1\}^{d_{\text{in}} \times m}$ and continuous codebook $\mathbf{c}^{(j)} \in \mathbb{R}^m$:

$$\begin{aligned} \underset{\substack{\mathbf{P}^{(j)} \in \{0,1\}^{d_{\text{in}} \times m} \\ \mathbf{c}^{(j)} \in \mathbb{R}^m}}{\text{minimize}} \quad & \sum_{j=1}^{d_{\text{out}}} \|\mathbf{X}\mathbf{w}_j - \mathbf{X}\mathbf{P}^{(j)}\mathbf{c}^{(j)}\|_2^2 \\ \text{subject to} \quad & \mathbf{P}^{(j)}\mathbf{1}_m = \mathbf{1}_{d_{\text{in}}}, \end{aligned}$$

LNQ: Algorithm

LNQ is a non-uniform scalar quantization method that alternates **closed-form codebook update** and **coordinate-descent assignment update**.

repeat:

$$\mathbf{c}^{(j)} \leftarrow \left(\mathbf{P}^{(j)\top}\mathbf{X}^\top\mathbf{X}\mathbf{P}^{(j)}\right)^{-1}\mathbf{P}^{(j)\top}\mathbf{X}^\top\mathbf{X}\mathbf{w}_j \qquad \text{(codebook)}$$

for $i = 1$ **to** d_{in} **do:**

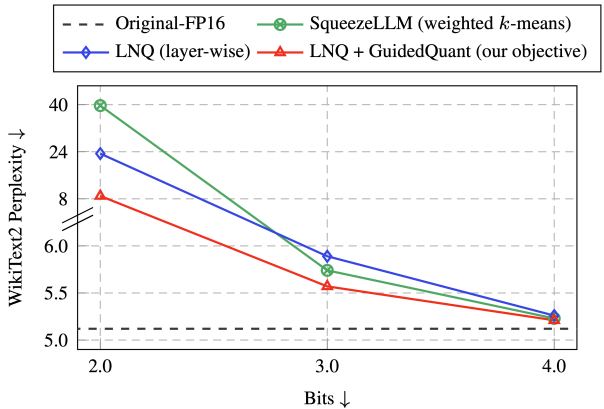
$$c_{q^*}^{(j)} \leftarrow \underset{\widehat{W}_{ij}\in\{c_1^{(j)},\ldots,c_m^{(j)}\}}{\operatorname{argmin}} \left(\widehat{\mathbf{w}}_j - \mathbf{w}_j\right)^\top\mathbf{X}^\top\mathbf{X}\left(\widehat{\mathbf{w}}_j - \mathbf{w}_j\right)$$

$$\forall q \in [m] : P_{iq}^{(j)} = \begin{cases} 1 & \text{if } q = q^*, \\ 0 & \text{otherwise.} \end{cases} \qquad \text{(assignment)}$$

$\rightarrow$ Monotonically decreases the objective and guarantees convergence.

LNQ: Results

LNQ algorithm is fully compatible with **GuidedQuant**: Together, they achieve state-of-the-art performance on **weight-only scalar quantization**.



Conclusion

- **GuidedQuant** integrates end loss gradients into the layer-wise quantization objective, outperforming PTQ methods.
- **LNQ** is a non-uniform scalar quantization method that alternates closed-form codebook update and coordinate-descent assignment update.

Code: `https://github.com/snu-mllab/GuidedQuant`