Gradient Enhancement Task Aware Post-training Quantization

Yihua Shao[1,2,3], **Yan Gu**[4], **Minxi Yan**[5], **Siyu Chen**[3], **Haiyang Liu**[3], **Ziyang Yan**[6], **Yongjia Li**[7], **Yan Wang**[8], **Qun Song**[9], **Hao Tang**[1*], **Haotong Qin**[2], **Jingcai Guo**[2*], **Nicu Sebe**[6]
[1]School of Computer Science, Peking University
[2]Department of Computing/LSGI, The Hong Kong Polytechnic University
[3]Institute of Automation, Chinese Academy of Sciences
[4]School of Advanced Manufacturing, Guangdong University of Technology
[5]Department of Computer Science and Engineering, Chinese University of Hong Kong
[6]Department of Information Engineering and Computer Science, University of Trento
[7]Department of Electrical and Electronic Engineering, HKUST
[8]Institute for AI Industry Research (AIR), Tsinghua University
[9]Department of Electrical Engineering, City University of Hong Kong
yihuajerry@gmail.com, haotang@pku.edu.cn, jc-jingcai.guo@polyu.edu.hk

Abstract

The huge parameters and extensive training corpora of Large Language Models (LLMs) empower them to tackle complex tasks. Yet, their massive size incurs substantial inference costs, hindering deployment on resource-constrained edge devices. Low-bit quantization provides a way to compress models and improve their practicality for edge-side deployment. Nonetheless, many edge applications do not require the full generalization abilities of LLMs and instead only depend on their knowledge in specific domains. This paper introduces Gradient Enhancement Task Aware Post-training Quantization, i.e., **GTAQ**, to address the generalization issue. Concretely, GTAQ locates task-critical weights via gradient-based validation and retrieval, and then amplifies these salient weights. For that purpose, GTAQ preserves and strengthens weights related to the target tasks, enabling task-aware uniform-bit quantization. We extensively evaluate the LLaMA family of language models on WikiText, C4, and MMLU. Our experiments show that GTAQ consistently delivers notable performance improvements across tasks, surpassing both general-purpose and task-aware quantization baselines. At the same time, GTAQ yields more than 3.5× speedup in model inference and substantially cuts storage requirements, while offering better optimization than mixed-precision quantization baselines. https://github.com/YihuaJerry/GTAQ.git.

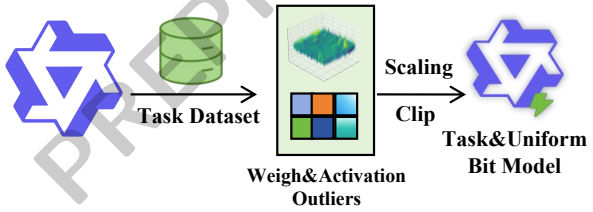

Figure 1: GTAQ obtains task-sensitive activations and weight outliers through a task-specialized dataset, followed by corresponding weight clipping and scaling, thereby achieving both task specialization and uniform-bit quantization of the model.

1 Introduction

Large language models (LLMs) [Wang *et al.*, 2026b; Achiam *et al.*, 2023; Almazrouei *et al.*, 2023; Touvron *et al.*, 2023] based on Transformer [Vaswani, 2017] have shown their outstanding capabilities in most of scenarios [Brown, 2020]. LLMs are capable of handling numerous complex tasks owing to their massive number of parameters [Chowdhery *et al.*, 2023; Wang *et al.*, 2026c] and vast amount of training data [Kaplan *et al.*, 2020; Wang *et al.*, 2026a]. However, the huge memory consumption leads to great difficulties in its application and deployment [Li *et al.*, 2024a] in resources limited devices. Therefore, when deploying large language models, there are often huge GPU clusters to support the inference of the models [Xu *et al.*, 2024]. Although LLMs possess strong generalization capabilities to adapt to diverse tasks, in edge systems, they only need to be optimized for a specific set of tasks rather than demonstrating full generality [Xiao *et al.*, 2025a]. To avoid retraining the model—as required by methods such as distillation [Hinton *et al.*, 2015] and fine-tuning [Shao *et al.*, 2025], post-training quantization (PTQ) becomes a more advantageous approach. Current PTQ methods can effectively reduce model inference costs and storage requirements, but the quantized models primarily focus on maintaining their generalization capability [Li *et al.*, 2024b; Huang *et al.*, 2024; Dettmers *et al.*, 2023; Frantar *et al.*, 2022]. This results in quantized models struggling to achieve satisfactory performance in certain complex tasks, such as

*Corresponding authors: *Hao Tang and Jingcai Guo*.

multilingual understanding[Frantar *et al.*, 2022]. Therefore, reducing the size of the model while ensuring its performance is necessary by design effective quantization algorithms [Lin *et al.*, 2024]. With effective quantization algorithms, researchers can make low-bit models' performance close to that of 16-bit models. For specific edge-task scenarios, designing quantization models that are both hardware-optimized and task-aligned is particularly imperative. Although current methods like TCAQ [Xiao *et al.*, 2025a] can retrieve task-specific circuits, its mixed-precision quantization still makes it difficult to deploy on edge systems.

To address the limitations of task-agnostic metrics, we propose a novel post-training quantization framework that reformulates the quantization objective as a task-aware optimization problem called Gradient-enhanced Weight Quantization (i.e., **GTAQ**). Diverging from conventional paradigms that minimize layer-wise reconstruction error in a task-agnostic manner, GTAQ explicitly aligns quantization noise with the sensitivity landscape of the downstream task. This proposed pipeline consists of two integral stages beginning with **Task-Specific Sensitivity Derivation**, where we employ a first-order Taylor expansion to rigorously quantify the marginal contribution of each parameter's perturbation to the global task loss, establishing a composite metric that integrates gradient information, proxy error estimation, and structural regularization. Subsequently, the **Importance-Guided Optimization** stage aggregates these element-wise sensitivities into channel-wise importance scores to guide a weighted reconstruction search. This adaptive process solves for quantization parameters that prioritize the fidelity of task-critical structures, effectively mitigating performance degradation.

We conducted extensive experiments on the Wikitext [Merity *et al.*, 2016], C4 [Raffel *et al.*, 2020], and MMLU [Hendrycks *et al.*, 2021] datasets to evaluate the capability of quantized models across different language tasks. Additionally, we measured the latency and storage consumption of the quantized models. The results show that GTAQ can enhance quantized models for corresponding tasks while achieving lower latency and a reduced storage footprint compared to other mixed-precision quantized models.

In summary, the contributions of this paper can be proposed as follows:

1) We propose a task-aware quantization method called Gradient-enhanced Weight Quantization (GTAQ), which identifies salient weights corresponding to specific tasks within LLMs through gradient-guided retrieval, thereby achieving enhanced performance on targeted tasks after quantization.
2) GTAQ achieves this by adaptively scaling the corresponding weight channels based on the identified salient weights. This enhances the targeted channels, enables uniform-bit quantization, and maintains hardware-friendly efficiency while preserving model capability.
3) Models quantized with GTAQ achieve state-of-the-art (SOTA) performance in both language modeling and massive multitask language understanding, outperforming both general-purpose and task-aware quantization methods, while also delivering over 3.5× speedup and significantly reduced storage footprint.

2 Related Works

2.1 Post-training Quantization

Post-training quantization (PTQ) applies mainly to visual models [Gholami *et al.*, 2022; Su *et al.*, 2026]. For example, AdaRound [Nagel *et al.*, 2020], AdaQuant [Hubara *et al.*, 2020], and BitSplit [Wang *et al.*, 2024b] can be used for models with less parameters. When models have a large number of parameters, they cause a significant loss but the reduction in memory is not significant. Most quantization methods are developed by Optimal Brain Quantization (OBQ) [Frantar and Alistarh, 2022]. OBQ is developed by the Optimal Brain Surgeon (OBS) [Hassibi *et al.*, 1993; Singh and Alistarh, 2020; Frantar *et al.*, 2021], which default a model response to input is 0 at the end of training. But the OBQ applies weight pruning to model quantization by quantifying the weights that have the less impact on the network. Some PTQ methods locating outlier by reinforcement learning [Wang *et al.*, 2024b] to retrieve the best outlier on a calibration set with a very large sample size. However, the reinforcement learning is not robust and is difficult to converge compared to other PTQ methods, making the calibration process inefficient.

2.2 Large Language Model Quantization

Currently, large language model quantization is mainly divided into quantization-based training [Liu *et al.*, 2023; Bondarenko *et al.*, 2024; Zhu *et al.*, 2023; Shao *et al.*, 2025] and post-training quantization. Quantization during training is very difficult to apply due to the large amount of computational resources and overhead required, as well as the large amount of training data. The remaining methods, such as ZeroQuant [Yao *et al.*, 2022], utilize knowledge distillation to train one transformer layer at a time instead of training all transformer layers directly. However, poor quality of certain data can also affect the performance of the model after quantization [Wang *et al.*, 2024a]. Post-training quantization requires less data compared to quantization-aware training, and only a few thousand or even a few hundred calibration sets are needed to complete the retrieval of sensitive weights for the model. This greatly reduces the data cost and the computational cost of model quantization. Both GPTQ [Frantar *et al.*, 2022] and SPQR [Dettmers *et al.*, 2023], quantization methods derived from the OBQ method, have shown superior performance in large language models. There are also methods such as AWQ [Lin *et al.*, 2024], LLM.int8() [Dettmers *et al.*, 2022] that determine the outlier of the model by searching for the activation of the model, and then quantify it with the corresponding method. For specific tasks, TCAQ [Xiao *et al.*, 2025a] achieves task-aware quantization through task circuits. However, TACQ employs mixed-precision quantization, which makes it difficult to deploy on edge devices. Therefore, we propose GTAQ, which achieves uniform-bit task-aware quantization.

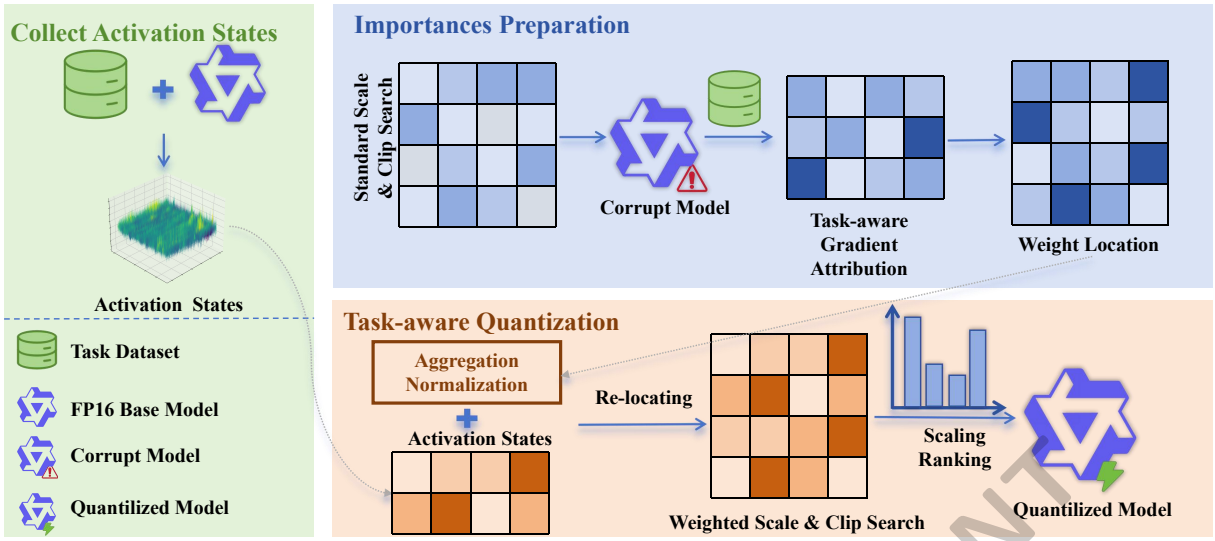

Figure 2: Pipeline of GTAQ. GTAQ first utilizes the corresponding task data as a validation set and collects gradients within the model through a frozen backpropagation process. Based on gradient variations, it locates outliers in the weights that are sensitive to this task. Simultaneously, based on the model's activations, it identifies the corresponding outlier weights. By normalizing the activation-correlated outlier weights with the task-correlated outlier weights and applying scaling, it achieves task-aware quantization for LLMs.

3 Methodology

In this section, we elaborate on the quantization process of GTAQ by addressing three key aspects: task definition, sensitive weight identification, and salient weight optimization.

3.1 Preliminaries and Problem Formulation

We focus on the uniform affine quantization for Linear layers in Large Language Models (LLMs). Let $\mathbf{W} \in \mathbb{R}^{C_{out} \times C_{in}}$ denote the full-precision weights. The quantized weights $\hat{\mathbf{W}}$ are obtained via a channel-wise quantizer $Q(\cdot)$:

$$\hat{\mathbf{W}} = Q(\mathbf{W}; \mathbf{s}, \mathbf{z}) = \mathbf{s} \odot \text{clip}\left(\lfloor \frac{\mathbf{W}}{\mathbf{s}} \rceil + \mathbf{z}, -2^{b-1}, 2^{b-1} - 1 \right) \tag{1}$$

where $\mathbf{s} \in \mathbb{R}^{C_{out}}$ is the scaling vector, $\mathbf{z}$ is the zero-point, and b is the bit-width.

Our primary objective is to find the optimal quantization parameters $\mathbf{s}^*$ that minimize the performance degradation on a specific downstream task dataset $\mathcal{D}_{task}$. This can be formulated as minimizing the expected task loss $\mathcal{L}_{task}$:

$$\mathbf{s}^* = \arg\min_{\mathbf{s}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{task}} \left[\mathcal{L}(\hat{\mathbf{W}}(\mathbf{s}); \mathbf{x}, y) \right]. \tag{2}$$

3.2 Task-Specific Sensitivity Derivation

Directly optimizing the discrete quantization parameters against the global loss function is intractable. Inspired by recent advances in gradient-based attribution [Xiao *et al.*, 2025b], we employ a **Taylor Series Approximation** to estimate the impact of quantization noise on the task loss.

First-Order Loss Approximation

Let $\Delta\mathbf{W} = \hat{\mathbf{W}} - \mathbf{W}$ be the perturbation introduced by quantization. We analyze the change in loss $\Delta\mathcal{L} = \mathcal{L}(\hat{\mathbf{W}}) - \mathcal{L}(\mathbf{W})$ using a second-order Taylor expansion around the original weights $\mathbf{W}$:

$$\Delta\mathcal{L} \approx \nabla_{\mathbf{W}}\mathcal{L}^T \Delta\mathbf{W} + \frac{1}{2}\Delta\mathbf{W}^T \mathbf{H} \Delta\mathbf{W} \tag{3}$$

where $\nabla_{\mathbf{W}}\mathcal{L}$ is the gradient and $\mathbf{H}$ is the Hessian matrix. Since computing the full Hessian is computationally prohibitive for LLMs, and assuming the model has converged to a local minimum where the gradient is small but non-negligible for task adaptation, we focus on the first-order term to capture the directional sensitivity of the loss landscape.

However, relying solely on the gradient $\nabla_{\mathbf{W}}\mathcal{L}$ (as in standard attribution) or solely on the quantization error $\Delta\mathbf{W}$ (as in Euclidean minimization) is insufficient. The former ignores the magnitude of the error, while the latter ignores the task relevance.

Composite Sensitivity Metric

To rigorously quantify the importance of each parameter, we construct a composite metric that bridges the gap between theoretical loss approximation and numerical stability. We define the element-wise sensitivity matrix $\mathbf{I} \in \mathbb{R}^{C_{out} \times C_{in}}$ as:

$$\mathbf{I}_{ij} = \underbrace{\left| \frac{\partial \mathcal{L}}{\partial \mathbf{W}_{ij}} \right|}_{\text{Task Criticality}} \cdot \underbrace{|\mathbf{W}_{ij} - Q_{std}(\mathbf{W}_{ij})|}_{\text{Quantization Noise}} \cdot \underbrace{|\mathbf{W}_{ij}|^{\gamma}}_{\text{Magnitude Prior}} \tag{4}$$

This formulation integrates three critical factors:

1. **Gradient Sensitivity:** The term $|\nabla_{\mathbf{W}}\mathcal{L}|$ aligns with the first-order Taylor term, indicating how steeply the task loss increases with respect to a perturbation in $\mathbf{W}_{ij}$.
2. **Proxy Error Estimation:** We use a standard quantization proxy $\mathbf{W}_{proxy} = Q_{std}(\mathbf{W})$ to simulate the expected perturbation magnitude $|\Delta\mathbf{W}_{ij}|$. This term ensures that weights susceptible to large rounding errors

(e.g., those near quantization boundaries) are penalized more heavily.

3. **Structural Regularization:** The term $|\mathbf{W}_{ij}|^{\gamma}$ (where typically $\gamma = 1$) acts as a structural prior. It incorporates the heuristic that larger magnitude weights often encode more significant features, preventing the metric from overfitting to noisy gradients in near-zero parameters.

Combining these, Eq. (4) provides a robust, task-aware upper bound on the expected loss degradation contribution of each weight.

3.3 Importance-Guided Optimization

Having derived the fine-grained sensitivity $\mathbf{I}$, we now transition to the optimization of the quantization parameters $\mathbf{s}$. Since quantization is applied per-channel, we aggregate the parameter-level importance into a channel-wise importance vector $\boldsymbol{\alpha} \in \mathbb{R}^{C_{out}}$.

Channel-wise Aggregation

We compute the cumulative sensitivity for the i-th output channel by summing the importance scores along the input dimension j:

$$\hat{\boldsymbol{\alpha}}_i = \sum_{j=1}^{C_{in}} \mathbf{I}_{ij} \tag{5}$$

To ensure numerical stability during the search phase, we apply a robust normalization:

$$\boldsymbol{\alpha}_i = \frac{\hat{\boldsymbol{\alpha}}_i - \mu_{\boldsymbol{\alpha}}}{\sigma_{\boldsymbol{\alpha}} + \epsilon} \tag{6}$$

where $\mu_{\boldsymbol{\alpha}}$ and $\sigma_{\boldsymbol{\alpha}}$ are the mean and standard deviation of the channel sensitivities.

Weighted Reconstruction Search

Finally, we reformulate the quantization objective. Instead of minimizing the raw Mean Squared Error (MSE) of the layer outputs (which treats all channels equally), we minimize the *Importance-Weighted Reconstruction Error*.

Let $\mathbf{X}_{cal} \in \mathbb{R}^{C_{in} \times N}$ be the activation input from the task-aligned calibration set. For each output channel i, we solve for the optimal step size s_i:

$$s_i^* = \underset{s_i \in \mathbb{R}^+}{\operatorname{argmin}} \ \boldsymbol{\alpha}_i \cdot \|\mathbf{W}_{i,:}\mathbf{X}_{cal} - Q(\mathbf{W}_{i,:}; s_i)\mathbf{X}_{cal}\|_2^2 \tag{7}$$

This objective effectively "reshapes" the optimization landscape. Channels with high task sensitivity (large $\boldsymbol{\alpha}_i$) dominate the loss, forcing the search algorithm (e.g., grid search or ternary search) to select an s_i that preserves the fidelity of these critical structures, potentially at the cost of larger errors in less sensitive channels.

4 Experiments

In this section, we experimentally validate GTAQ's adaptability across different models and its generalization capability across various tasks, while also discussing the core factors influencing GTAQ's performance.

4.1 Experiments Setting

Setups. We selected Llama2-7B and 13B [Touvron *et al.*, 2023] and Llama3-8B [Dubey *et al.*, 2024] for the experiments. To evaluate our method, we pick standard datasets that cover a variety of model use scenarios. As for language modeling tasks, we choose C4 [Raffel *et al.*, 2020] and WikiText2 [Merity *et al.*, 2016] datasets. As for the massive multitask language understanding task, we utilize the MMLU [Hendrycks *et al.*, 2021] dataset to extensively validate the feasibility of our method. All experiments were conducted on a single NVIDIA A800 GPU.

Baselines. For task-aware methods, we select TACQ [Xiao *et al.*, 2025a] as our baseline. For general-purpose quantization methods, we select GPTQ [Frantar *et al.*, 2022], SLiM [Huang *et al.*, 2024], SqueezeLLM [Kim *et al.*, 2023] and SPQR [Dettmers *et al.*, 2023] as our baselines.

Evaluation Metrics. For the language modeling task, we focus on the model's perplexity (PPL↓) and the next token prediction accuracy (Acc↑). The model's perplexity is expressed in Eq. 8:

$$\text{PPL} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N} \log P(x_i \mid x_{0:i-1})\right), \tag{8}$$

where x_i represents the predicted token conditioned on the former tokens ($x_{<i}$) in inference step i.

The next token prediction accuracy is defined as:

$$\text{Acc} = \frac{N_{\hat{y}_t = \overrightarrow{y_t}}}{T}, \tag{9}$$

where $\hat{y}_t$ is the predicted token by LLMs, $\overrightarrow{y_t}$ is the ground truth token, and T is the total number of tokens.

4.2 Main Results

Massive Multitask Language Results. Compared to language modeling tasks, massive multitask language processing requires models to possess more sophisticated scene understanding capabilities. Therefore, employing uniform quantization can impair the performance of the compressed model. As shown in Table 1, on the MMLU dataset, GTAQ significantly outperforms the task-aware quantization baseline TACQ. This indicates that within the framework of task-aware quantization, GTAQ can mitigate the quantization loss in corresponding weight circuits. Furthermore, GTAQ achieves a lower average bit-width than TACQ, thereby significantly enhancing performance and alleviating the loss associated with task-aware quantization. Compared to other baseline methods, GTAQ quantization also demonstrates significant performance improvement across different tasks. Therefore, in complex language modeling tasks, GTAQ not only achieves task-specific optimization of model performance but also reduces overall model degradation. In both simple language modeling tasks and complex task understanding, models quantized with GTAQ show significant improvement over baselines. Thus, GTAQ-quantized models can adapt to different tasks, demonstrating their task-agnostic nature. Furthermore, GTAQ requires less validation data compared to baseline methods, thus allowing for some

Model	Methods	Avg Bit	Calib. Num	Accuracy (%) Average	STEM	Humanities	Social Sciences
Llama-2-7B	Original	16.00	-	45.30	36.40	42.90	51.20
	GPTQ [Frantar *et al.*, 2022]	4.00	1024	41.47	32.08	38.53	46.18
	SLiM [Huang *et al.*, 2024]	4.00	1024	43.24	34.52	40.76	48.53
	TACQ [Xiao *et al.*, 2025a]	4.10	128	44.06	35.18	41.63	49.82
	GTAQ (Ours)	**4.00**	**128**	**44.82**	**35.87**	**42.38**	**50.57**
	GPTQ [Frantar *et al.*, 2022]	3.00	1024	34.23	26.47	32.08	38.42
	SLiM [Huang *et al.*, 2024]	3.00	1024	38.54	30.12	36.38	42.83
	Squeeze [Kim *et al.*, 2023]	3.15	1024	36.82	28.53	34.12	40.57
	SPQR [Dettmers *et al.*, 2023]	3.14	1024	35.95	27.82	33.25	39.88
	TACQ [Xiao *et al.*, 2025a]	3.12	128	40.18	31.77	38.24	45.08
	GTAQ (Ours)	**3.00**	**128**	**41.53**	**32.86**	**39.52**	**46.47**
	GPTQ [Frantar *et al.*, 2022]	2.00	1024	25.82	22.07	24.53	26.18
	Squeeze [Kim *et al.*, 2023]	2.15	1024	25.33	21.84	24.08	25.82
	SPQR [Dettmers *et al.*, 2023]	2.14	1024	24.96	21.52	23.75	25.42
	TACQ [Xiao *et al.*, 2025a]	2.10	128	29.54	24.76	28.12	31.48
	GTAQ (Ours)	**2.00**	**128**	**32.06**	**26.53**	**30.38**	**34.23**
Llama-2-13B	Original	16.00	-	54.80	44.10	52.80	62.60
	GPTQ [Frantar *et al.*, 2022]	4.00	1024	51.53	40.78	49.23	59.07
	SLiM [Huang *et al.*, 2024]	4.00	1024	53.08	42.53	51.12	60.84
	TACQ [Xiao *et al.*, 2025a]	4.10	128	53.87	43.18	51.93	61.47
	GTAQ (Ours)	**4.00**	**128**	**54.38**	**43.76**	**52.42**	**62.08**
	GPTQ [Frantar *et al.*, 2022]	3.00	1024	46.47	36.23	44.52	53.08
	SLiM [Huang *et al.*, 2024]	3.00	1024	49.83	39.47	47.82	56.53
	Squeeze [Kim *et al.*, 2023]	3.15	1024	48.48	38.25	46.52	55.08
	SPQR [Dettmers *et al.*, 2023]	3.14	1024	47.81	37.53	45.88	54.27
	TACQ [Xiao *et al.*, 2025a]	3.12	128	51.18	40.82	49.47	58.23
	GTAQ (Ours)	**3.00**	**128**	**52.06**	**41.53**	**50.38**	**59.12**
	GPTQ [Frantar *et al.*, 2022]	2.00	1024	28.53	24.08	27.23	30.47
	Squeeze [Kim *et al.*, 2023]	2.15	1024	28.04	23.82	27.05	29.84
	SPQR [Dettmers *et al.*, 2023]	2.14	1024	27.67	23.53	26.68	29.27
	TACQ [Xiao *et al.*, 2025a]	2.10	128	36.78	30.23	35.07	40.52
	GTAQ (Ours)	**2.00**	**128**	**39.17**	**32.53**	**37.76**	**43.08**
Llama-3-8B	Original	16.00	-	66.51	54.88	63.42	75.74
	GPTQ [Frantar *et al.*, 2022]	3.00	1024	56.30	47.14	51.20	64.17
	SLiM [Huang *et al.*, 2024]	3.125	1024	61.95	51.36	58.54	71.40
	Squeeze [Kim *et al.*, 2023]	3.15	1024	59.66	51.14	57.01	69.25
	SPQR [Dettmers *et al.*, 2023]	3.14	1024	58.65	49.78	56.17	66.15
	TACQ [Xiao *et al.*, 2025a]	3.12	128	63.28	52.42	60.54	71.66
	GTAQ (Ours)	**3.00**	**128**	**64.15**	**53.15**	**61.40**	**72.30**
	GPTQ [Frantar *et al.*, 2022]	2.00	1024	26.76	27.84	26.08	26.06
	SLiM [Huang *et al.*, 2024]	2.125	1024	34.84	32.94	35.31	40.99
	Squeeze [Kim *et al.*, 2023]	2.15	1024	26.27	27.92	26.62	26.02
	SPQR [Dettmers *et al.*, 2023]	2.14	1024	25.90	23.88	23.77	25.85
	TACQ [Xiao *et al.*, 2025a]	2.10	128	49.19	41.12	47.98	56.34
	GTAQ (Ours)	**2.00**	**128**	**50.24**	**42.05**	**48.66**	**57.12**

Table 1: Results on MMLU Dataset. In subtasks, GTAQ achieves state-of-the-art (SOTA) performance compared to both general-purpose quantization methods and task-aware quantization methods. Furthermore, GTAQ requires less validation data to perform more effective calibration. Also, GTAQ achieves SOTA performance, demonstrating its applicability to all tasks.

reduction in validation cost, and can also help mitigate privacy risks associated with data collection.

Language Modeling Results. As shown in Table 2, when the model is quantized to 3-bit, the GTAQ-quantized model exhibits less performance degradation in language modeling tasks compared to the current SOTA method, TACQ-quantized models. This indicates that GTAQ mitigates the structural damage to the model caused by TACQ during quantization. Compared to other general quantization methods, GTAQ also demonstrates significant improvements when using the same calibration set. This suggests that GTAQ effectively enhances the task adaptation capability of quantized models. Even when the model is quantized to 2-bit, the GTAQ-quantized model still shows significant performance improvement compared to models quantized by other methods. This demonstrates that GTAQ not only optimizes the quantized model for specific tasks but also effectively preserves non-outlier weights. Additionally, the GTAQ-quantized model has a lower average bit-width compared to the baselines, indicating that GTAQ imposes a lower impact on the model than other methods.

Efficiency Comparison. To evaluate the efficiency of the quantized model, we measured the latency and storage when the model was quantized to 3 bits. As shown in Table 3, compared to TACQ, GTAQ achieves higher throughput with a lower storage footprint. By quantizing models to a uniform bit-width, GTAQ significantly improves throughput while reducing storage requirements compared to mixed-precision quantization methods. Therefore, compared to other baseline methods, GTAQ is more hardware-friendly and more suitable for deployment on edge devices. However, compared to GPTQ, GTAQ only approaches its throughput level. Because GPTQ compensates for outliers in the final layer, the overall weight distribution of GTAQ is sparser than that of GPTQ,

Method	Avg Bit	Llama-2-7B WikiText2	Llama-2-7B C4	Llama-2-13B WikiText2	Llama-2-13B C4	Llama-3-8B WikiText2	Llama-3-8B C4
Original	16.00	5.47 / 59.32	6.97 / 52.18	4.88 / 61.24	6.47 / 54.56	6.17 / 58.42	8.99 / 51.71
GPTQ [Frantar *et al.*, 2022]	2.00	512.47 / 0.12	186.23 / 0.85	325.82 / 1.17	142.93 / 2.48	410.63 / 0.00	136.62 / 0.00
SLiM [Huang *et al.*, 2024]	2.125	148.23 / 4.17	94.38 / 5.82	112.53 / 6.47	82.08 / 8.83	248.06 / 0.50	96.58 / 4.10
Squeeze [Kim *et al.*, 2023]	2.15	405.23 / 0.50	155.28 / 1.20	305.15 / 1.50	128.45 / 3.15	195.39 / 0.00	287.54 / 0.00
SPQR [Dettmers *et al.*, 2023]	2.14	485.67 / 0.20	175.45 / 0.95	318.52 / 1.25	138.67 / 2.80	5.95e5 / 0.00	4.14e5 / 0.00
TACQ [Xiao *et al.*, 2025a]	2.10	12.53 / 30.47	16.78 / 28.23	10.18 / 34.52	14.53 / 32.08	17.66 / 36.50	25.19 / 33.15
GTAQ (Ours)	**2.00**	**10.82 / 32.08**	**14.18 / 30.53**	**8.87 / 36.23**	**12.38 / 34.52**	**15.88 / 37.20**	**22.45 / 34.90**
GPTQ [Frantar *et al.*, 2022]	3.00	9.82 / 35.48	12.53 / 30.18	8.48 / 38.23	10.82 / 34.47	12.28 / 39.54	16.41 / 32.45
SLiM [Huang *et al.*, 2024]	3.125	8.18 / 38.23	10.52 / 34.47	7.18 / 41.53	9.23 / 38.18	8.15 / 44.20	12.53 / 38.50
Squeeze [Kim *et al.*, 2023]	3.15	8.85 / 37.05	11.25 / 33.15	7.82 / 40.15	9.85 / 36.85	7.93 / 45.10	12.85 / 38.20
SPQR [Dettmers *et al.*, 2023]	3.14	9.25 / 36.25	11.85 / 31.85	8.15 / 39.25	10.32 / 35.52	9.28 / 42.10	14.38 / 37.80
TACQ [Xiao *et al.*, 2025a]	3.10	7.47 / 40.53	9.18 / 36.82	6.53 / 44.18	8.08 / 40.82	7.53 / 46.80	11.81 / 39.20
GTAQ (Ours)	**3.00**	**6.78 / 42.13**	**8.47 / 38.53**	**5.88 / 46.52**	**7.52 / 42.78**	**7.12 / 48.85**	**10.95 / 42.05**

Table 2: Results on Language Modeling Tasks. When quantized to 3 bits, GTAQ achieves state-of-the-art (SOTA) results compared to other quantization methods while maintaining a lower average bit-width. When quantized to 2 bits, GTAQ still maintains competitive performance, indicating its adaptability to lower bit-width regimes.

	Average bit	Calibration number	Llama-2-7B-hf Speedup (↑)	Llama-2-7B-hf Memory (↓)	Llama-2-13B-hf Speedup (↑)	Llama-2-13B-hf Memory (↓)	Llama-3-8B-hf Speedup (↑)	Llama-3-8B-hf Memory (↓)
Original	16	-	×1.00	12.83G	×1.00	23.63G	×1.00	14.57G
GPTQ [Frantar *et al.*, 2022]	3.00	1024	×**3.89**	2.55G	×**3.74**	4.65G	×**3.86**	2.85G
SLiM [Huang *et al.*, 2024]	3.125	1024	×2.95	2.70G	×2.98	4.88G	×2.92	3.05G
Squeeze [Kim *et al.*, 2023]	3.05	128	×2.25	2.65G	×2.30	4.80G	×2.20	2.95G
SPQR [Dettmers *et al.*, 2023]	3.35	128	×1.28	2.90G	×1.35	5.25G	×1.25	3.25G
TACQ [Xiao *et al.*, 2025a]	3.10	128	×2.76	2.62G	×2.79	4.75G	×2.73	2.95G
GTAQ (Ours)	**3.00**	**128**	×3.45	**2.55G**	×3.48	**4.65G**	×3.42	**2.85G**

Table 3: Results on Efficiency Comparison. We quantized the entire model to 3 bits. The quantized models produced by GTAQ demonstrate significantly higher throughput and lower storage requirements compared to task-aware methods. This indicates that GTAQ is more hardware-friendly and better suited for deployment on edge devices.

making it comparatively less hardware-friendly. When compared to other post-training quantization methods, GTAQ also achieves more significant acceleration effects and requires less storage, demonstrating its hardware-friendly nature.

4.3 Ablation Study

In this section, we conduct ablation studies on individual components of the algorithm, including the size of the validation set and the optimization module, to investigate the impact of each factor on GTAQ.

Effect of Different Components. To evaluate the individual contributions of the components constituting our proposed sensitivity metric, we conduct a component-wise ablation study. We assess four different configurations to verify the necessity of incorporating gradient signals ($|\nabla\mathcal{L}|$), weight magnitudes ($|W|$), and quantization error estimates ($|W - W_{proxy}|$).

The variants are formalized as follows:

- **Weight Magnitude ($|W|$):** A heuristic relying solely on the absolute value of weights, disregarding task-specific information.
- **Gradient Sensitivity ($|\nabla\mathcal{L}|$):** Utilizes the magnitude of the task-specific gradient as the sole indicator of parameter importance.
- **Grad × Mag:** A composite metric combining structural weight magnitude and task gradient, defined as $|\nabla\mathcal{L}| \cdot |W|$.
- **GTAQ (Ours):** The proposed full formulation $\alpha = |\nabla\mathcal{L}| \cdot |W| \cdot |W - W_{proxy}|$, which further weighs parameters based on their inherent susceptibility to quantization noise.

Sensitivity Metric (α)	Formulation	2-bit	3-bit
Weight Magnitude	$\|W\|$	32.50	59.80
Gradient Sensitivity	$\|\nabla\mathcal{L}\|$	35.10	61.20
Grad × Mag	$\|\nabla\mathcal{L}\| \cdot \|W\|$	46.80	63.45
GTAQ (Ours)	$\|\nabla\mathcal{L}\| \cdot \|W\| \cdot \|W - W_{proxy}\|$	**50.24**	**64.15**

Table 4: Quantitative analysis of the sensitivity metric components. We report the MMLU accuracy (%) under 2-bit and 3-bit quantization settings. The results demonstrate that the proposed metric, which accounts for quantization noise sensitivity, yields superior performance compared to partial baselines.

Analysis. The quantitative results presented in Table 4 reveal several critical insights regarding the quantization objective:

First, the baseline utilizing only *Weight Magnitude* exhibits the lowest performance, particularly in the 2-bit regime

(32.50%). This corroborates that preserving high-magnitude weights alone is insufficient for maintaining downstream task reasoning capabilities.

Second, while *Gradient Sensitivity* incorporates task-aware signals, its standalone performance is unstable. This instability is likely attributed to the stochastic nature of gradients in parameters with negligible magnitudes, which can mislead the scale optimization process.

Third, the integration of gradient information with weight magnitude (*Grad* $\times$ *Mag*) yields a substantial improvement, increasing 2-bit accuracy from $\sim$35% to 46.80%. This validates the hypothesis that effective quantization requires a synergy between task relevance (gradient) and structural priors (magnitude).

Finally, the full **GTAQ** metric achieves the optimal performance across all settings. By explicitly incorporating the quantization error term $|W - W_{proxy}|$, the metric effectively penalizes parameters that are numerically fragile to quantization perturbations. This indicates that awareness of quantization noise is a prerequisite for robust low-bit adaptation.

Impact of Optimization Objective To validate the effectiveness of our proposed *Importance-Weighted Reconstruction* (Eq. 7), we compare it against the standard Mean Squared Error (MSE) minimization objective during the 2-bit quantization of LLaMA3-7b. Standard MSE assigns equal weight to all channels ($\boldsymbol{\alpha}_i = 1$), while our method weights the reconstruction error by the derived channel sensitivity.

Optimization Objective	PPL ↓	MMLU ↑
Standard MSE ($\boldsymbol{\alpha}_i = 1$)	25.12	38.45
Weighted MSE (Ours)	**22.45**	**50.24**

Table 5: Ablation on the Optimization Objective. We compared standard MSE against weight importance-based MSE by quantizing LLaMA3 to 2 bits. The results indicate that our designed weight importance-based MSE method achieves superior performance.

As shown in Table 5, the use of a weighted objective function leads to significant performance improvements. This confirms that prioritizing the reconstruction of task-critical channels identified by our metric can effectively preserve downstream task performance, even when the global per-channel MSE may be slightly higher. Hence, Importance-Weighted Reconstruction enables superior performance of the quantized model.

Effect on Calibration Set Size. To investigate the impact of different validation set sizes on GTAQ, we varied the number of validation samples under 2-bit quantization and conducted tests on the WikiText and MMLU datasets.

As shown in Table 6, the results indicate that with a validation set size of 128, the model quantized by GTAQ achieves superior performance compared to using other validation set sizes. This demonstrates that GTAQ achieves optimal quantization quality when calibrated with 128 validation samples. Meanwhile, GTAQ exhibits relatively stable performance across different validation set sizes. This suggests that the model can achieve good performance with GTAQ processing across various validation sets.It can also consistently deliver robust results across a variety of tasks.

Calibration Samples	PPL ↓	MMLU ↑
1	23.97	48.76
32	24.10	48.90
64	23.05	49.80
128 (Our)	22.45	50.24
256	22.53	50.11
512	22.42	50.20

Table 6: Abolation on Calibration Set Size. GTAQ maintains largely consistent performance of the quantized model across different validation set sizes, with slightly more prominent results when using 128 validation samples.

Calib. \ Infer.	WikiText2	C4
WikiText2	**6.78**	8.95
C4	7.15	**8.47**

Table 7: Impact of Calibration Domain Alignment.

Effect on Calibration Domain. As illustrated in Table 7, we utilized Llama-2-7B quantized to 3 bits specifically optimized for either WikiText2 or C4, and evaluated them cross-wise. The results demonstrate a clear pattern: the model achieves the lowest perplexity when the calibration dataset aligns with the inference domain. specifically, the model calibrated on WikiText2 achieves a PPL of 6.78 on WikiText2, outperforming the model calibrated on C4 (7.15). Similarly, using C4 for calibration yields superior performance on the C4 evaluation set compared to using WikiText2 (8.47 vs. 8.95).

5 Conclusions

In this paper, we propose Gradient Enhancement Task Aware Post-training Quantization (GTAQ). GTAQ employs frozen backpropagation to obtain gradients from LLMs, thereby identifying task-sensitive weights. By scaling the task-sensitive weights, GTAQ enhances their impact. The results demonstrate that GTAQ consistently achieves state-of-the-art (SOTA) performance on the LLaMA family of models in both language modeling and massive multitask language understanding tasks, outperforming both general-purpose quantization methods and existing task-aware quantization approaches. This demonstrates that models quantized with GTAQ exhibit generalization capability across different model scales and achieve enhanced performance across diverse tasks. Simultaneously, GTAQ achieves over 3.5× speedup and significantly reduces the storage. Therefore, GTAQ enhances quantized models for specific tasks, providing a solution for task-aware model customization.

Acknowledgements

This work was supported in part by the Hong Kong RGC General Research Fund (Grant Nos. 15221123, 15216424, and 15211525), the Hong Kong PolyU Internal Research

Fund (Grant Nos. P0058468 and P0056171), and the Fundamental Research Funds for the Central Universities (Peking University).

References

[Achiam *et al.*, 2023] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[Almazrouei *et al.*, 2023] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*, 2023.

[Bondarenko *et al.*, 2024] Yelysei Bondarenko, Riccardo Del Chiaro, and Markus Nagel. Low-rank quantization-aware training for llms. *arXiv preprint arXiv:2406.06385*, 2024.

[Brown, 2020] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[Chowdhery *et al.*, 2023] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[Dettmers *et al.*, 2022] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35:30318–30332, 2022.

[Dettmers *et al.*, 2023] Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. Spqr: A sparse-quantized representation for near-lossless llm weight compression. *arXiv preprint arXiv:2306.03078*, 2023.

[Dubey *et al.*, 2024] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

[Frantar *et al.*, 2021] Elias Frantar, Eldar Kurtic, and Dan Alistarh. M-fac: Efficient matrix-free approximations of second-order information. *Advances in Neural Information Processing Systems*, 34:14873–14886, 2021.

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

[Gholami *et al.*, 2022] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*, pages 291–326. Chapman and Hall/CRC, 2022.

[Hassibi *et al.*, 1993] Babak Hassibi, David G Stork, and Gregory J Wolff. Optimal brain surgeon and general network pruning. In *IEEE international conference on neural networks*, pages 293–299. IEEE, 1993.

[Hendrycks *et al.*, 2021] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2021.

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

[Huang *et al.*, 2024] Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Xianglong Liu, Luca Benini, Michele Magno, and Xiaojuan Qi. Slim-llm: Salience-driven mixed-precision quantization for large language models. *arXiv preprint arXiv:2405.14917*, 2024.

[Hubara *et al.*, 2020] Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. Improving post training neural quantization: Layer-wise calibration and integer programming. *arXiv preprint arXiv:2006.10518*, 2020.

[Kaplan *et al.*, 2020] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[Kim *et al.*, 2023] Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W Mahoney, and Kurt Keutzer. Squeezellm: Dense-and-sparse quantization. *arXiv preprint arXiv:2306.07629*, 2023.

[Li *et al.*, 2024a] Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, et al. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459*, 2024.

[Li *et al.*, 2024b] Zhengang Li, Alec Lu, Yanyue Xie, Zhenglun Kong, Mengshu Sun, Hao Tang, Zhong Jia Xue, Peiyan Dong, Caiwen Ding, Yanzhi Wang, et al. Quasar-vit: Hardware-oriented quantization-aware architecture search for vision transformers. In *Proceedings of the 38th ACM International Conference on Supercomputing*, pages 324–337, 2024.

[Lin *et al.*, 2024] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm

compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100, 2024.

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

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

[Nagel *et al.*, 2020] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, pages 7197–7206. PMLR, 2020.

[Raffel *et al.*, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[Shao *et al.*, 2025] Yihua Shao, Minxi Yan, Yang Liu, Siyu Chen, Wenjie Chen, Xinwei Long, Ziyang Yan, Lei Li, Chenyu Zhang, Nicu Sebe, et al. In-context meta lora generation. *arXiv preprint arXiv:2501.17635*, 2025.

[Singh and Alistarh, 2020] Sidak Pal Singh and Dan Alistarh. Woodfisher: Efficient second-order approximation for neural network compression. *Advances in Neural Information Processing Systems*, 33:18098–18109, 2020.

[Su *et al.*, 2026] Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel space reasoning via curiosity-driven reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[Touvron *et al.*, 2023] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[Vaswani, 2017] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[Wang *et al.*, 2024a] Yifei Wang, Jizhe Zhang, and Yisen Wang. Do generated data always help contrastive learning? *arXiv preprint arXiv:2403.12448*, 2024.

[Wang *et al.*, 2024b] Yingchun Wang, Song Guo, Jingcai Guo, Yuanhong Zhang, Weizhan Zhang, Qinghua Zheng, and Jie Zhang. Data quality-aware mixed-precision quantization via hybrid reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[Wang *et al.*, 2026a] Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *Advances in Neural Information Processing Systems*, 38:30865–30891, 2026.

[Wang *et al.*, 2026b] Haozhe Wang, Haoran Que, Qixin Xu, Minghao Liu, Wangchunshu Zhou, Jiazhan Feng, Wanjun Zhong, Wei Ye, Tong Yang, Wenhao Huang, Ge Zhang, and Fangzhen Lin. Reverse-engineered reasoning for open-ended generation. In *The Fourteenth International Conference on Learning Representations*, 2026.

[Wang *et al.*, 2026c] Haozhe Wang, Qixin Xu, Che Liu, Junhong Wu, Fangzhen Lin, and Wenhu Chen. Emergent hierarchical reasoning in LLMs through reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

[Xiao *et al.*, 2025a] Hanqi Xiao, Yi-Lin Sung, Elias Stengel-Eskin, and Mohit Bansal. Task-circuit quantization: Leveraging knowledge localization and interpretability for compression. In *Second Conference on Language Modeling*, 2025.

[Xiao *et al.*, 2025b] Hanqi Xiao, Yi-Lin Sung, Elias Stengel-Eskin, and Mohit Bansal. Task-circuit quantization: Leveraging knowledge localization and interpretability for compression, 2025.

[Xu *et al.*, 2024] Mengwei Xu, Wangsong Yin, Dongqi Cai, Rongjie Yi, Daliang Xu, Qipeng Wang, Bingyang Wu, Yihao Zhao, Chen Yang, Shihe Wang, et al. A survey of resource-efficient llm and multimodal foundation models. *arXiv preprint arXiv:2401.08092*, 2024.

[Yao *et al.*, 2022] Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in Neural Information Processing Systems*, 35:27168–27183, 2022.

[Zhu *et al.*, 2023] Xuekai Zhu, Biqing Qi, Kaiyan Zhang, Xinwei Long, Zhouhan Lin, and Bowen Zhou. Pad: Program-aided distillation can teach small models reasoning better than chain-of-thought fine-tuning. *arXiv preprint arXiv:2305.13888*, 2023.