

HEAL: Latent Head Alignment for Post-Training Quantization of Large Language Models

Mingzhe Yang^{1*}, Sihao Lin^{2*}, Ruibo Zhang^{1*}, Xinyuan Zhu³,
Lei Chen¹, Weidong Chen¹, Xiaojun Chang^{1†},

¹ University of Science and Technology of China,

² Adelaide University, ³ Ant Group

yangmz@mail.ustc.edu.cn, xjchang@ustc.edu.cn

Abstract

Post-training quantization (PTQ) has become a key approach for large language model (LLM) compression. Existing methods commonly adopt block-level reconstruction as a surrogate to calibrate each quantized Transformer block. However, our analysis shows that this loose objective can admit *cancellation patterns*, where the aggregated block-level reconstruction error is small even though the per-head errors remain large. We therefore revisit the PTQ formulation from a variational perspective. By treating per-head representations as latent variables and the block output as the observed variable, we find that conventional block-level reconstruction corresponds to optimizing only the observation-level term, while omitting a latent alignment term that captures head-level quantization error. Motivated by this finding, we propose **HEAL**, a Latent **HE**ad **AL**ignment framework that turns block-level reconstruction calibration into joint observation-latent calibration with stop-gradient head alignment. Experiments on LLaMA-2/3 and Qwen2.5 families under low-bit weight-activation quantization show that HEAL generally improves perplexity and zero-shot accuracy over block-level reconstruction baselines.

1 Introduction

The practical deployment of large language models (LLMs) is increasingly constrained by memory footprint and inference cost (Zhao et al., 2026; Zhu et al., 2024; Zhou et al., 2024; Wan et al., 2024; Kwon et al., 2023). Low-bit weight-activation post-training quantization (PTQ) has emerged as a key compression technique for LLMs, lowering the numerical precision of both stored weights and run-time activations after pre-training (Gholami et al., 2022; Sun et al., 2025; Zhao et al., 2024; Shao et al., 2024). To reduce the quantization error, existing

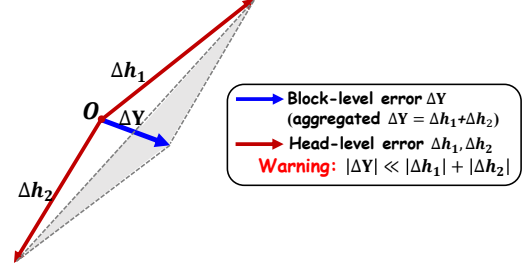


Figure 1: Illustration of hidden head-wise quantization errors under block-level reconstruction. Although the aggregated output error $\Delta Y = \Delta h_1 + \Delta h_2$ is small due to error cancellation between different heads, the individual head errors $|\Delta h_1|$ and $|\Delta h_2|$ are still large. This indicates that block-level reconstruction alone can fail to constrain head-level representation distortion.

PTQ methods introduce various smoothing transformations, such as scaling, rotation, or learnable affine maps (Sun et al., 2025; Ashkboos et al., 2024; Liu et al., 2025; Shao et al., 2024; Lin et al., 2024a). However, directly optimizing these transformations against the end-to-end quantization error is computationally prohibitive at LLM scale. Some prior work therefore adopts block-level reconstruction as a tractable surrogate, calibrating each quantized Transformer block to match the output of its full-precision counterpart.

This surrogate, however, observes head-wise quantization errors only after the head representations have been linearly mixed and absorbed into the block output. As a result, it provides a loose signal for preserving per-head representations. That is, head-wise quantization errors pointing in different directions can partially cancel after projection, yielding a small aggregated error that nonetheless masks substantial per-head distortion. For instance, Figure 1 illustrates this issue with a toy example, where two heads with large but opposing error vectors produce a near-zero block-level error.

We further investigate this problem from an analytical perspective (see Section 3.1). A first-order expansion shows that the block-level reconstruction

*Equal contribution.

†Corresponding author.

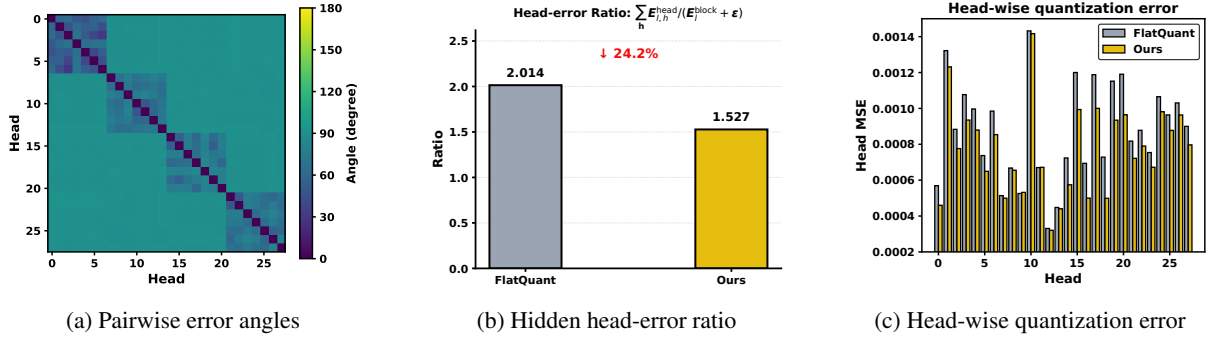


Figure 2: Analysis of hidden head-wise quantization errors. (a) Pairwise angles between head-wise quantization error vectors, showing that errors from different heads can have different directions. (b) Hidden head-error ratio, measured by $\sum_h E_{l,h}^{\text{head}} / (E_l^{\text{block}} + \epsilon)$. A larger ratio indicates that the aggregated block-level error underestimates the accumulated head-level distortions. Compared with FlatQuant, our method reduces this ratio by 24.2%, suggesting that head-wise errors are less hidden after calibration. (c) Head-wise quantization error comparison between FlatQuant and our method. Our method generally reduces the errors of individual attention heads, supporting the hypothesis that HEAL mitigates hidden head errors associated with cancellation.

loss can be approximated by the norm of a sum of head-wise error contributions. This sum-of-vectors structure can admit *cancellation patterns*, where the block-level error is small even though per-head quantization errors remain large. Such patterns are problematic because block-level reconstruction cannot distinguish genuine head-wise error reduction from cancellation-based fits. We empirically demonstrate this phenomenon on Qwen2.5-7B in Figure 2: (a) head-wise quantization error vectors span a wide range of pairwise angles, confirming that cancellation is geometrically available; (b) the ratio $\sum_h E_{l,h}^{\text{head}} / (E_l^{\text{block}} + \epsilon)$ remains large under block-level calibration, indicating that block-level error systematically underestimates accumulated head-level distortion.

To address this gap, we revisit PTQ calibration through a surrogate variational view. By treating per-head representations as latent variables and the full-precision block output as the observed variable, we show that the MSE-based PTQ formulation can be interpreted as a negative evidence lower bound (ELBO) that decomposes into an observation-level reconstruction term and a latent-alignment term. Conventional block-level reconstruction recovers only the first term, leaving the latent head alignment unconstrained and thereby admitting cancellation patterns.

Motivated by this finding, we propose **HEAL**, a plug-and-play latent **HE**ad **AL**ignment framework that turns block-level reconstruction calibration into joint observation-latent calibration, without changing the quantizer or inference graph. To make the latent alignment practical at LLM scale,

we further introduce a stochastic token-window calibration strategy that computes the head-alignment loss on randomly sampled continuous token windows, reducing the additional computation required by head alignment while exposing the objective to diverse local contexts. Experiments on LLaMA-2/3 and Qwen2.5 families under low-bit weight-activation quantization show that HEAL generally improves perplexity and zero-shot accuracy over block-level reconstruction baselines. Our contributions are summarized as follows:

- We identify *cancellation patterns* in block-level reconstruction. Head-wise quantization errors can cancel after projection, making block-level error unable to distinguish genuine head-wise error reduction.
- We derive an evidence lower bound (ELBO) for PTQ calibration, showing that existing block-level reconstruction losses omit the latent head alignment term.
- We propose **HEAL**, a plug-and-play latent head-alignment framework that complements block-level reconstruction with a stop-gradient head alignment objective and stochastic token-window calibration, improving perplexity and zero-shot accuracy across various LLM backbones and downstream tasks.

2 Related Work

2.1 LLM Quantization.

Quantization compresses model parameters by representing full-precision values with low-bit formats,

thereby reducing storage overhead and inference-time memory consumption. In general, existing quantization methods can be categorized into two main paradigms: Quantization-Aware Training (QAT) and Post-Training Quantization (PTQ).

QAT (Liu et al., 2024a; Chen et al., 2024; Jeon et al., 2025a,b; Ke et al., 2024) improves low-bit robustness by exposing LLMs to quantization effects during training process. However, this benefit comes at the cost of additional model adaptation, which is often prohibitively expensive for LLMs with billions of parameters. PTQ methods (Xiao et al., 2023; Lin et al., 2024b; Huang et al., 2024a; Wang et al., 2026; Kim et al., 2025; Liu et al., 2024b) instead quantize pre-trained LLMs directly, thus substantially reducing the computational cost of quantization. SliderQuant (Wang et al., 2026) proposes inter-layer and intra-layer sliding quantization, which adaptively reduces layer-wise quantization errors through a sliding-window mechanism. GuidedQuant (Kim et al., 2025) introduces end-loss guidance into the PTQ objective by incorporating gradient information during calibration.

2.2 Transformation LLM Quantization.

Recent LLM PTQ methods have mainly focused on mitigating outliers and improving the quantizability of weights and activations through equivalent transformations. OmniQuant (Shao et al., 2024) introduces a differentiable PTQ framework with learnable weight clipping and learnable equivalent transformation, enabling both weight-only and weight-activation quantization under a block-wise reconstruction objective. QuaRot (Ashkboos et al., 2024) applies computationally invariant rotations to remove activation outliers and enables end-to-end 4-bit quantization of weights, activations, and KV caches. SpinQuant (Liu et al., 2025) further improves rotation-based PTQ by learning the rotation matrices, since different random rotations can lead to large variations in quantized model accuracy. DuQuant (Lin et al., 2024a) redistributes massive and normal activation outliers via block-wise rotation and zigzag permutation. FlatQuant (Sun et al., 2025) proposes learnable affine transformations to enhance the flatness of weights and activations.

3 Method

We first analyze the limitation of block-level reconstruction for low-bit quantization (Section 3.1). We then revisit PTQ calibration from a variational

perspective and identify a missing latent head-alignment term. Based on this analysis, we introduce **HEAL** in Section 3.2, which reformulates block-level reconstruction calibration as joint observation-latent calibration with stop-gradient head alignment (Section 3.4). The overview of the proposed method is shown in Figure 3.

3.1 The Devil in the Details: Cancellation in Block-level Reconstruction

Let $\mathbf{H}_{l,h}^{fp}$ and $\mathbf{H}_{l,h}^q$ denote the full-precision and quantized representations of the h -th attention head in the l -th Transformer block before the attention output projection. We define the corresponding head-wise quantization error as

$$\Delta \mathbf{H}_{l,h} = \mathbf{H}_{l,h}^{fp} - \mathbf{H}_{l,h}^q. \quad (1)$$

Since the output projection linearly mixes different heads, block-level reconstruction observes these head-wise errors only after they have been projected and combined.

Proposition 1. Under a first-order approximation of the post-attention block transformation, the block-level reconstruction loss can be approximated as

$$\mathcal{L}_{\text{rec}} \approx \left\| \sum_{h=1}^{N_h} \mathbf{J}_l (\Delta \mathbf{H}_{l,h} \mathbf{W}_{o,l,h}) + \Delta \mathbf{Z}_l^{\text{mlp}} \right\|_F^2, \quad (2)$$

where $\mathbf{W}_{o,l,h}$ is the output-projection submatrix associated with the h -th head, $\mathbf{J}_l$ is the local Jacobian of the post-attention transformation, and $\Delta \mathbf{Z}_l^{\text{mlp}}$ denotes errors from the MLP branch and activation perturbations. Consequently, there exist nonzero head-wise quantization errors for which $\mathcal{L}_{\text{rec}}$ is small or even zero while the accumulated per-head error remains large.

Discussion. Proposition 1 shows that block-level reconstruction optimizes a mixed sum of projected head-wise errors rather than the errors of individual heads. When different heads induce error contributions with opposing directions, the mixed block-level error can be small due to cancellation. For example, in a simplified two-head case with $\mathbf{J}_l = \mathbf{I}$, $\mathbf{W}_{o,l,1} = \mathbf{W}_{o,l,2} = \mathbf{\Gamma}$, $\Delta \mathbf{Z}_l^{\text{mlp}} = \mathbf{0}$, $\Delta \mathbf{H}_{l,1} = \mathbf{E}$, and $\Delta \mathbf{H}_{l,2} = -\mathbf{E}$, the block-level reconstruction error becomes zero while the accumulated head-wise error is $2\|\mathbf{E}\|_F^2$. Thus, block-level reconstruction cannot distinguish genuine head-wise error reduction from cancellation-based fits. The proof is provided in Appendix A.

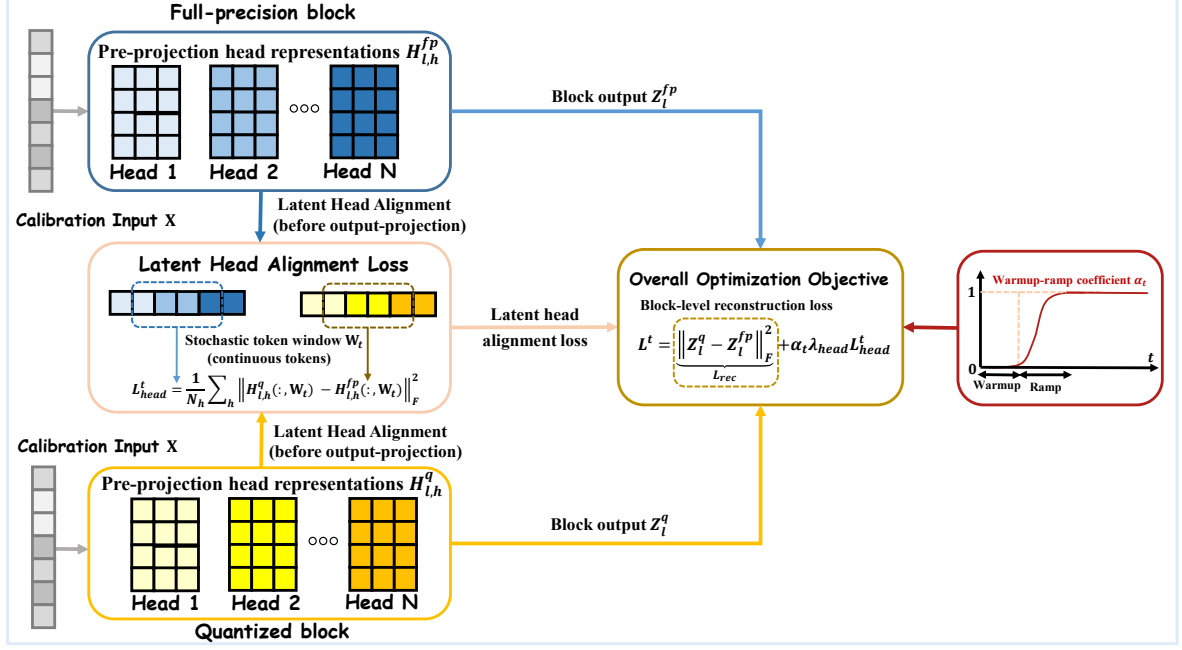


Figure 3: Overview of HEAL. HEAL complements block-level reconstruction with a latent head-aware calibration objective. Given the same calibration input, HEAL aligns the full-precision and quantized pre-projection attention-head representations, in addition to matching the final block outputs. The head alignment loss is computed on a stochastic token window and gradually introduced by a Warmup-Ramp coefficient.

3.2 Latent Head Alignment

The cancellation analysis above shows that block-level reconstruction can be satisfied without reducing the underlying head-wise errors. Instead of directly adding a heuristic head-level regularization term, we revisit the PTQ formulation from a variational perspective.

Proposition 2. We treat pre-projection attention-head representations as latent variables and the full-precision block output as the observed target. Under a surrogate Gaussian interpretation of MSE-based PTQ calibration, the negative evidence lower bound (ELBO) can be written, up to constants and variance-dependent coefficients, as

$$\mathcal{L}_{\text{var}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{head}} \sum_{h=1}^{N_h} \left\| \mathbf{H}_{l,h}^q - \mathbf{H}_{l,h}^{fp} \right\|_F^2 + C, \quad (3)$$

where $\mathcal{L}_{\text{rec}} = \|\mathbf{Z}_l^q - \mathbf{Z}_l^{fp}\|_F^2$, and C is independent of the calibration parameters. This view shows that conventional block-level reconstruction corresponds to retaining only the observation-level term $\mathcal{L}_{\text{rec}}$, leaving latent head alignment unconstrained.

Discussion. Proposition 2 shows that the head-alignment term is not an ad-hoc regularizer, but a latent alignment term omitted by block-level reconstruction. This missing term directly targets the

head-wise quantization errors that can be hidden by cancellation in the mixed block output. The proof is provided in Appendix B.

3.3 HEAL as a Plug-and-Play Calibration Objective

Motivated by the above discussion, we propose **HEAL**, a plug-and-play calibration objective that turns block-level PTQ calibration into joint observation-latent calibration by explicitly aligning pre-projection head representations. It can be incorporated into block-level reconstruction-based PTQ pipelines that expose pre-projection attention-head representations.

For the h -th attention head, the representation discrepancy is measured as

$$\mathcal{E}_{l,h} = \frac{1}{|\Omega_{l,h}|} \left\| \mathbf{H}_{l,h}^q - \mathbf{H}_{l,h}^{fp} \right\|_F^2, \quad (4)$$

where $\mathcal{E}_{l,h}$ is the normalized discrepancy and $\Omega_{l,h}$ denotes all elements of the h -th attention head. The head-wise latent alignment loss is then defined as

$$\mathcal{L}_{\text{head}} = \frac{1}{N_h} \sum_{h=1}^{N_h} \mathcal{E}_{l,h}, \quad (5)$$

This objective explicitly enforces latent representation consistency for each attention head by aligning

the pre-projection head representations of the full-precision and quantized models. Meanwhile, the stop-gradient operation detaches the full-precision representations from the computational graph, making them fixed reference targets and preventing gradients from propagating through the full-precision branch.

3.4 Stochastic Token-window Calibration

The head-level constraint to all tokens can over-constrain the original block-level reconstruction process. Therefore, we adopt a stochastic token-window strategy. In the process, the head-wise latent alignment loss is evaluated only on a sampled continuous token window. At calibration step t , we sample a window start index from the valid token positions:

$$s_t \sim \text{Uniform}\{0, 1, \dots, S - L\}, \quad (6)$$

where S denotes sequence length, and construct a continuous token window of length L :

$$\mathcal{W}_t = \{s_t, s_t + 1, \dots, s_t + L - 1\}, \quad (7)$$

where L is the token-window size. For the h -th attention head, the windowed representation discrepancy is computed as:

$$\mathcal{E}_{l,h}^{(t)} = \frac{1}{|\Omega_{l,h}^{(t)}|} \left\| \mathbf{H}_{l,h}^q(:, \mathcal{W}_t) - \mathbf{H}_{l,h}^{fp}(:, \mathcal{W}_t) \right\|_F^2, \quad (8)$$

where $\Omega_{l,h}^{(t)}$ denotes all elements of the h -th attention head within the sampled token window. Therefore, the head-wise latent alignment loss using stochastic token-window can be defined as:

$$\mathcal{L}_{\text{head}}^{(t)} = \frac{1}{N_h} \sum_{h=1}^{N_h} \mathcal{E}_{l,h}^{(t)}. \quad (9)$$

Across calibration steps, different token windows are sampled, exposing the head-alignment objective to diverse local token contexts. This stochastic strategy avoids forcing the objective to fit the entire sequence at every optimization step.

3.5 Optimization Objective

In the calibration process, to avoid disturbing the early stage of block-level reconstruction, we introduce a warmup-ramp coefficient to gradually activate the head-alignment loss:

$$\alpha_t = \begin{cases} 0, & t < T_{\text{warm}}, \\ \frac{t - T_{\text{warm}}}{T_{\text{ramp}}}, & T_{\text{warm}} \leq t < T_{\text{warm}} + T_{\text{ramp}}, \\ 1, & t \geq T_{\text{warm}} + T_{\text{ramp}}, \end{cases} \quad (10)$$

Algorithm 1 Latent Head Alignment Process

Require: Full-precision block f_l^{fp} , quantized block f_l^q , calibration inputs $\mathbf{X}$, calibration steps T

- 1: Initialize calibration parameters θ_l for f_l^q
- 2: Cache full-precision block outputs $\mathbf{Z}_l^{fp}$ and pre-projection head representations $\mathbf{H}_l^{fp}$
- 3: **for** $t = 1$ to T **do**
- 4: Run quantized forward with θ_l to obtain $\mathbf{Z}_l^q$ and $\mathbf{H}_l^q$
- 5: Compute block reconstruction loss $\mathcal{L}_{\text{rec}}^{(t)}$
- 6: Sample a continuous token window $\mathcal{W}_t$
- 7: Compute windowed head-wise discrepancies $\{\mathcal{E}_{l,h}^{(t)}\}_{h=1}^{N_h}$ between $\mathbf{H}_l^{fp}$ and $\mathbf{H}_l^q$
- 8: Compute head alignment loss $\mathcal{L}_{\text{head}}^{(t)} = \frac{1}{N_h} \sum_{h=1}^{N_h} \mathcal{E}_{l,h}^{(t)}$
- 9: Compute total calibration loss $\mathcal{L}^{(t)} = \mathcal{L}_{\text{rec}}^{(t)} + \alpha_t \lambda_{\text{head}} \mathcal{L}_{\text{head}}^{(t)}$
- 10: Update calibration parameters θ_l by minimizing $\mathcal{L}^{(t)}$
- 11: **end for**
- 12: **return** Calibrated parameters θ_l

where T_{warm} and T_{ramp} denote the warmup and ramp-up stages, respectively. Therefore, at calibration step t , the final optimization objective can be expressed as:

$$\mathcal{L}^{(t)} = \mathcal{L}_{\text{rec}} + \alpha_t \lambda_{\text{head}} \mathcal{L}_{\text{head}}^{(t)}, \quad (11)$$

where $\mathcal{L}_{\text{rec}}$ is the conventional block-level reconstruction loss, λ_{head} controls the strength of the head alignment, and α_t gradually increases the contribution of the head-wise latent alignment loss during calibration.

Importantly, HEAL performs block-wise PTQ calibration and only augments the calibration objective with the proposed head-wise latent alignment loss and optimizes the learnable calibration parameters θ_l of each quantized block, such as affine transformation and quantizer clipping/scaling, not introducing additional learnable modules. The full-precision block is frozen and used only to generate detached targets $\mathbf{Z}_l^{fp}$ and $\mathbf{H}_l^{fp}$. The quantized block is calibrated by jointly minimizing the reconstruction loss and the proposed head-wise latent alignment loss. We summarize the overall calibration procedure in Algorithm 1.

MODEL	METHOD	WIKITEXT2	ARC-C	ARC-E	BOOLQ	HELLASWAG	PIQA	WINOGRANDE	AVG
LLAMA2-7B	FP16	5.47	46.25	74.58	77.68	76.01	79.11	69.06	70.45
	OMNIQUANT	16.02	25.43	46.89	59.97	39.02	62.79	50.43	47.42
	QUAROT	8.38	36.26	63.43	69.14	65.45	73.12	60.06	61.24
	DUQUANT	6.26	36.86	63.72	67.65	69.34	74.70	63.38	62.61
	SPINQUANT	6.08	39.33	66.96	69.42	71.70	74.76	66.06	64.71
	FLATQUANT	5.76	44.71	71.93	74.83	74.36	77.80	67.48	68.52
	OURS	5.75	42.75	73.91	74.59	74.30	77.97	67.80	68.55
LLAMA2-13B	FP16	4.88	49.06	77.53	80.55	79.39	80.52	72.06	73.19
	OMNIQUANT	14.84	25.76	49.20	61.99	40.72	65.89	53.91	49.58
	QUAROT	6.17	44.62	71.51	74.43	72.36	76.99	65.51	67.57
	DUQUANT	5.36	41.98	70.00	70.24	73.95	77.20	66.93	66.72
	SPINQUANT	5.48	45.99	73.40	78.84	74.93	77.45	69.93	70.09
	FLATQUANT	5.09	46.84	76.01	80.46	76.85	78.76	70.96	71.65
	OURS	5.09	46.93	75.84	80.40	77.80	78.67	70.80	71.74
LLAMA3.1-8B	FP16	6.24	53.92	81.14	82.20	78.97	81.12	73.09	75.07
	QUAROT	10.41	38.91	66.92	72.35	66.59	74.05	64.56	63.90
	SPINQUANT	8.69	43.69	70.41	78.81	73.93	78.24	63.18	68.04
	FLATQUANT	6.94	50.77	77.65	81.01	76.46	79.38	71.43	72.78
	OURS	6.94	52.13	78.16	80.70	76.97	80.20	72.38	73.42
QWEN2.5-7B	FP16	6.85	51.11	77.44	84.68	78.93	79.71	72.77	74.11
	QUAROT	10.97	44.11	69.53	65.50	71.44	76.99	63.77	65.22
	SPINQUANT	8.44	46.93	73.70	78.04	72.74	77.37	65.04	68.97
	FLATQUANT	8.17	52.47	77.69	83.79	76.30	78.35	69.38	73.00
	OURS	7.29	52.22	78.96	83.85	77.38	79.00	70.24	73.61
QWEN2.5-14B	FP16	5.29	59.04	79.34	85.60	83.00	82.05	75.30	77.39
	QUAROT	7.83	51.79	76.30	74.89	74.81	78.78	66.30	70.48
	SPINQUANT	7.52	53.50	79.80	84.16	73.84	78.51	70.40	73.37
	FLATQUANT	5.97	56.14	80.51	85.44	76.30	81.01	74.90	75.72
	OURS	5.97	57.00	81.23	84.46	81.63	81.39	75.53	76.87
QWEN2.5-32B	FP16	5.02	56.14	77.95	87.55	84.02	82.21	75.45	77.22
	QUAROT	6.67	48.98	75.25	82.39	78.03	79.87	67.64	72.03
	SPINQUANT	6.19	52.99	78.03	84.01	78.33	80.58	73.65	74.60
	FLATQUANT	5.45	53.84	77.61	85.32	83.38	81.01	75.53	76.12
	OURS	5.45	55.55	77.82	86.21	83.38	81.39	76.16	76.75

Table 1: The PPL and zero-shot performance of LLaMA2/3 and Qwen2.5 families under W4A4 setting. Bold results highlight the best performance.

4 Experiments

4.1 Implementation Details

Experimental Setup. We evaluate different methods on representative LLMs from the LLaMA-2/3 (Touvron et al., 2023; Grattafiori et al., 2024) and Qwen2.5 (Qwen et al., 2025) families using the Hugging Face Transformers library. All experiments are conducted on NVIDIA A100 GPUs with 80 GB memory. For all HEAL-enhanced variants, we use the same calibration data, calibration steps, quantization bit-widths, and optimization schedule as the corresponding base PTQ method. For example, in FlatQuant-based experiments, the optimized parameters include affine transformation parameters, diagonal scaling factors, and learnable clipping factors. We use block-wise calibration with a sequence length of 2048, a calibration batch size of 4, and $T = 15$ optimization steps for each Transformer block, which are the same as FlatQuant. For HEAL, the token-window length is set to $L = 1024$, the latent head alignment weight

λ_{head} is set to 0.003 for LLaMA-2/3 models and 0.002 for Qwen2.5 models (See Appendix D for analysis). We set the warm-up steps to $T_{\text{warm}} = 3$ and the ramp-up steps to $T_{\text{ramp}} = 5$ in all experiments.

Baselines. We compare HEAL with representative quantization methods, including OmniQuant (Shao et al., 2024), QuaRot (Ashkboos et al., 2024), DuQuant (Lin et al., 2024a), SpinQuant (Liu et al., 2025), and FlatQuant (Sun et al., 2025). Some of these methods calibrate quantization-related parameters or transformations using block-level output reconstruction objectives, without explicitly accounting for head-wise pre-projection quantization errors. Unless otherwise specified, HEAL is instantiated by adding our latent head-alignment objective to FlatQuant under the same quantizer and calibration setting. For a controlled comparison, we first evaluate all baselines under the W4A4 (both weights and activations are quantized to 4 bits) setting with the same RTN quantizer in Ta-

ble 1. We further report results under different bit-width settings and quantizers to examine the generality of HEAL.

Datasets and Evaluation Metrics. The 128 samples from the Wikitext2 (Merity et al., 2016) training dataset are randomly selected as the calibration set. We report perplexity (PPL) on Wikitext2 datasets and evaluate downstream zero-shot performance using standard tasks, including PIQA (Bisk et al., 2019); WinoGrande (Sakaguchi et al., 2021); HellaSwag (Zellers et al., 2019); ARC-e and ARC-c (Clark et al., 2018); BoolQ (Clark et al., 2019). Besides, we report few-shot results on ReCoRD (Zhang et al., 2018), SocialIQA (Sap et al., 2019), CommonsenseQA (Talmor et al., 2019), and MMLU (Hendrycks et al., 2020). All downstream evaluations are conducted using the lm_eval framework.

4.2 Main Results

Table 1 reports the W4A4 quantization performance of different quantization methods on the LLaMA-2/3 and Qwen2.5 model families. As shown in the table, OmniQuant suffers from noticeable performance degradation under the W4A4 setting, whereas recent transformation-based PTQ methods, including QuaRot, DuQuant, SpinQuant, and FlatQuant, substantially improve low-bit quantization performance. Among them, FlatQuant serves as a strong baseline and achieves competitive average performance across multiple LLMs. Building on this strong baseline, our method further improves the average zero-shot performance on most evaluated LLMs. In particular, on Qwen2.5-32B, our method matches or outperforms FlatQuant on all zero-shot datasets and increases the average score of zero-shot tasks. These results demonstrate that the proposed head-aware calibration objective can effectively enhance existing block-level reconstruction-based PTQ methods.

4.3 Analysis under Different Bit-widths

To evaluate the effectiveness of HEAL under different quantization bits, we conduct experiments on LLaMA3.1-8B with three representative weight-activation bit-width settings, including W2A4, W3A3, and W5A5. As shown in Figure 4, HEAL improves the average zero-shot accuracy across all bit-width settings. The improvement is more pronounced under the most challenging W2A4 setting. Nevertheless, HEAL improves the average zero-shot accuracy from 35.45% to 36.34%. When the

bit-width increases, the quantization error is relatively smaller, leading to more moderate but still consistent improvements. These results demonstrate that our method remains effective across different weight-activation bit-widths.

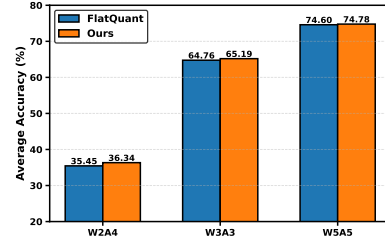


Figure 4: The results under different bit-widths.

4.4 Combination with Existing PTQ Methods

To verify the compatibility of HEAL with existing PTQ methods, we integrate it into two representative transformation-based quantization methods, QuaRot and SpinQuant. As shown in Table 2, adding HEAL improves the corresponding baselines across different model families under low-bit weight-activation quantization. Specifically, for QuaRot, integrating HEAL brings the largest improvements on LLaMA3.1-8B in both zero-shot accuracy and Wikitext2 perplexity. These results demonstrate that HEAL is complementary to existing quantization approaches. Instead of modifying the quantizer or transformation design, HEAL refines the calibration objective from a latent head alignment perspective and provides fine-grained supervision for attention-head representations. Therefore, our method can serve as a plug-in calibration objective to enhance the performance of different PTQ frameworks. The detailed results for zero-shot datasets are provided in Table 5.

MODEL	METHOD	WIKITEXT2	ACCURACY
LLAMA2-7B	QUAROT	8.38	61.24
	QUAROT+HEAL	7.90	62.11
	SPINQUANT	6.08	64.71
	SPINQUANT+HEAL	6.03	65.24
LLAMA2-13B	QUAROT	6.17	67.57
	QUAROT+HEAL	5.97	68.78
	SPINQUANT	5.48	70.09
	SPINQUANT+HEAL	5.38	70.26
LLAMA3.1-8B	QUAROT	10.41	63.90
	QUAROT+HEAL	9.02	67.66
	SPINQUANT	8.69	68.04
	SPINQUANT+HEAL	8.48	69.40

Table 2: The results of combining our method with other quantization methods.

4.5 Analysis on GPTQ Quantizer

We further adopt GPTQ (Frantar et al., 2022) as the weight quantizer to examine whether HEAL is compatible with other quantizers. As shown in

the Table 3, HEAL achieves comparable or lower PPL than FlatQuant, while improving the average zero-shot accuracy on LLaMA-2/3 and Qwen2.5 families. Notably, the Qwen2.5-14B model quantized by our method even achieves higher zero-shot accuracy than the Qwen2.5-32B models quantized by QuaRot and SpinQuant. These results demonstrate that the proposed HEAL is compatible with the GPTQ quantizer. The detailed results for each zero-shot datasets are provided in Table 7.

MODEL	METHOD	WIKITEXT2	ACCURACY
LLAMA2-7B	QUAROT	6.14	67.15
	SPINQUANT	5.89	67.32
	FLATQUANT	5.73	68.79
	OURS	5.73	69.33
LLAMA2-13B	QUAROT	5.40	70.05
	SPINQUANT	5.21	71.45
	FLATQUANT	5.07	71.48
	OURS	5.07	72.13
LLAMA3.1-8B	QUAROT	8.42	68.32
	SPINQUANT	7.38	69.48
	FLATQUANT	6.86	72.91
	OURS	6.85	73.37
QWEN2.5-7B	QUAROT	9.58	67.19
	SPINQUANT	7.86	71.47
	FLATQUANT	7.95	72.68
	OURS	7.62	73.44
QWEN2.5-14B	QUAROT	7.10	71.71
	SPINQUANT	6.87	73.62
	FLATQUANT	5.90	76.80
	OURS	5.89	76.87
QWEN2.5-32B	QUAROT	6.26	74.60
	SPINQUANT	6.19	73.69
	FLATQUANT	5.40	77.51
	OURS	5.40	77.67

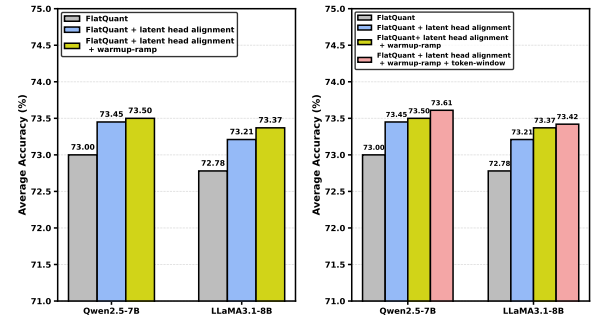
Table 3: The PPL and zero-shot results of using GPTQ.

4.6 Ablation Studies

Effect of Warmup-Ramp. Figure 5a reports the ablation study of the warmup-ramp schedule. Compared with directly applying the head-alignment objective, using warmup-ramp brings consistent improvements on Qwen2.5-7B and LLaMA3.1-8B. These results suggest that gradually introducing the head-level constraint can stabilize calibration and avoid over-constraining the block-level reconstruction objective at the early optimization stage.

Effect of Token-window Calibration. After validating the effectiveness of Warmup-Ramp, we further conduct an ablation analysis on token-window calibration under a fixed Warmup-Ramp setting. Specifically, all compared settings use the same Warmup-Ramp strategy, and differ only in whether token-window is applied. Compared with applying head-level alignment to all tokens, the results shown in Figure 5b demonstrate token-window can alleviate excessive constraints on block-level reconstruction, thereby leading to more stable and

reliable zero-shot performance improvements.



(a) Effect of warmup-ramp.

(b) Effect of token-window.

Figure 5: The ablation studies of warmup-ramp and token-window calibration.

Different Calibration Dataset. In the quantization process, using different calibration datasets can affect the performance of the quantized LLM. Therefore, we should evaluate the robustness of PTQ methods using different calibration datasets. We use C4 (Raffel et al., 2020) training datasets as the other calibration dataset and quantize Qwen2.5-7B under W4A4 setting. The quantized model is evaluated on the C4 validation dataset and the zero-shot task datasets. As shown in Figure 6, our method consistently outperforms FlatQuant.

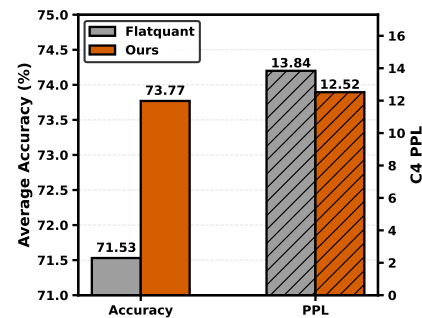


Figure 6: The influence of different calibration datasets.

Comparison with other alignment targets. We compare HEAL with two alternative alignment targets: attention-output alignment and MLP-state alignment. As shown in Table 4, latent head alignment achieves the best performance, indicating that explicitly constraining individual attention heads is more effective than aligning aggregated attention outputs and generic MLP states.

METHOD	QWEN2.5-7B	LLAMA3.1-8B
HEAD ALIGNMENT	73.61	73.42
ATTENTION-OUTPUT ALIGNMENT	72.67	72.78
MLP-STATE ALIGNMENT	72.92	72.91

Table 4: Comparison with other alignment targets.

5 Conclusion

In this paper, we propose HEAL, a latent head alignment framework for PTQ LLMs. Motivated

by the observation that block-level reconstruction can hide per-head quantization errors, we revisit PTQ calibration from a variational perspective and identify the missing latent head-alignment term in existing reconstruction objectives. HEAL addresses this gap by reformulating block-level reconstruction calibration as joint observation-latent calibration with a stop-gradient head-alignment objective and an efficient stochastic token-window strategy. The experiment results show that HEAL improves the performance over block-level reconstruction-based baselines. Our findings highlight the importance of preserving latent head-level representations.

Limitations

The paper is mainly evaluated on representative LLMs under standard PTQ settings. Therefore, the conclusions of this work are primarily supported by these evaluated model families and quantization configurations. Its behavior on broader architectures, larger-scale models, longer-context scenarios remains less extensively explored.

Acknowledgments

This work was partially supported by the New Generation Artificial Intelligence National Science and Technology Major Project (2025ZD0123100) and the National Natural Science Foundation of China (NSFC) under Grant Nos. 62573399 and U25A20530.

References

- Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. 2019. Variational information distillation for knowledge transfer. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9155–9163. IEEE.
- Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefer, and James Hensman. 2024. Quarot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 37:100213–100240.
- Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. [Piqa: Reasoning about physical commonsense in natural language](#). *Preprint*, arXiv:1911.11641.
- Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. 2024. Efficientqat: Efficient quantization-aware training for large language models. *arXiv preprint arXiv:2407.11062*.
- Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.
- Elias Frantar, Saleh Ashkboos, Torsten Hoefer, and Dan Alistarh. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.
- Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2022. A survey of quantization methods for efficient neural network inference. In *Low-power computer vision*, pages 291–326. Chapman and Hall/CRC.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and Xiaojuan Qi. 2024a. Slim-llm: Saliency-driven mixed-precision quantization for large language models. *arXiv preprint arXiv:2405.14917*.
- Yaomin Huang, Zaomin Yan, Chaomin Shen, Faming Fang, and Guixu Zhang. 2024b. Harmonizing knowledge transfer in neural network with unified distillation. In *European Conference on Computer Vision*, pages 58–74. Springer.
- Hyesung Jeon, Yulhwa Kim, and Jae-Joon Kim. 2025a. L4q: Parameter efficient quantization-aware fine-tuning on large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2002–2024.
- Hyesung Jeon, Seojune Lee, Beomseok Kang, Yulhwa Kim, and Jae-Joon Kim. 2025b. Qwha: Quantization-aware walsh-hadamard adaptation for parameter-efficient fine-tuning on large language models. *arXiv preprint arXiv:2509.17428*.

- Wenjing Ke, Zhe Li, Dong Li, Lu Tian, and Emad Barsoum. 2024. DI-qat: Weight-decomposed low-rank quantization-aware training for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 113–119.
- Jinuk Kim, Marwa El Halabi, Wonpyo Park, Clemens JS Schaefer, Deokjae Lee, Yeonhong Park, Jae W Lee, and Hyun Oh Song. 2025. Guidedquant: Large language model quantization via exploiting end loss guidance. *arXiv preprint arXiv:2505.07004*.
- Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.
- Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. 2024a. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. *Advances in Neural Information Processing Systems*, 37:87766–87800.
- Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024b. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of machine learning and systems*, 6:87–100.
- Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2024a. Llm-qat: Data-free quantization aware training for large language models. In *Findings of the association for computational linguistics: ACL 2024*, pages 467–484.
- Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. 2025. Spinquant: Llm quantization with learned rotations. In *International Conference on Learning Representations*, volume 2025, pages 92009–92032.
- Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024b. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *arXiv preprint arXiv:2402.02750*.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavathula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.
- Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 4463–4473.
- Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. 2024. [Omniquant: Omnidirectionally calibrated quantization for large language models](#). *Preprint*, arXiv:2308.13137.
- Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, Xin Jiang, Wulong Liu, and Jun Yao. 2025. [Flatquant: Flatness matters for llm quantization](#). *Preprint*, arXiv:2410.09426.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.
- 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, and 49 others. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *Preprint*, arXiv:2307.09288.
- Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, Mosharaf Chowdhury, and Mi Zhang. 2024. [Efficient large language models: A survey](#). *Preprint*, arXiv:2312.03863.
- Shigeng Wang, Chao Li, Yangyuxuan Kang, Jiawei Fan, Zhonghong Ou, and Anbang Yao. 2026. Sliderquant:

Accurate post-training quantization for llms. *arXiv preprint arXiv:2603.25284*.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International conference on machine learning*, pages 38087–38099. PMLR.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Zican Dong, Yupeng Hou, Beichen Zhang, Yingqian Min, Junjie Zhang, Peiyu Liu, Xiaolei Wang, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, and 4 others. 2026. [A survey of large language models](#). *Frontiers of Computer Science*, 20(12).

Yilong Zhao, Chien-Yu Lin, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind Krishnamurthy, Tianqi Chen, and Baris Kasikci. 2024. Atom: Low-bit quantization for efficient and accurate llm serving. *Proceedings of Machine Learning and Systems*, 6:196–209.

Zixuan Zhou, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Luning Wang, Zhihang Yuan, Xiuhong Li, Shengen Yan, Guohao Dai, Xiao-Ping Zhang, Yuhao Dong, and Yu Wang. 2024. [A survey on efficient inference for large language models](#). *Preprint*, arXiv:2404.14294.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2024. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577.

A Proof of Proposition 1

Let $\mathbf{H}_{l,h}^{fp} \in \mathbf{R}^{BS \times d_h}$ denote the full-precision representation of the h -th attention head in the l -th Transformer block before the attention output projection, where B is the batch size, S is the sequence length, and d_h is the head dimension. The concatenated multi-head representation is

$$\mathbf{H}_l^{fp} = \text{Concat}(\mathbf{H}_{l,1}^{fp}, \mathbf{H}_{l,2}^{fp}, \dots, \mathbf{H}_{l,N_h}^{fp}), \quad (12)$$

where N_h is the number of attention heads. The attention output before subsequent block operations is

$$\mathbf{A}_l^{fp} = \mathbf{H}_l^{fp} \mathbf{W}_{o,l}. \quad (13)$$

Similarly, let $\mathbf{H}_{l,h}^q$ denote the quantized head representation, and define the head-wise quantization error as

$$\Delta \mathbf{H}_{l,h} = \mathbf{H}_{l,h}^{fp} - \mathbf{H}_{l,h}^q. \quad (14)$$

Since multi-head attention outputs are concatenated along the head dimension before the output projection, the output projection matrix can be partitioned according to head subspaces:

$$\mathbf{W}_{o,l} = [\mathbf{W}_{o,l,1}, \mathbf{W}_{o,l,2}, \dots, \mathbf{W}_{o,l,N_h}]^T. \quad (15)$$

Therefore, the attention-output error induced by head-wise errors can be written as

$$\Delta \mathbf{A}_l = \sum_{h=1}^{N_h} \Delta \mathbf{H}_{l,h} \mathbf{W}_{o,l,h}. \quad (16)$$

Let $\mathcal{G}_l(\cdot)$ denote the post-attention operations within the block, including residual connections, normalization, and MLP transformations. The full-precision block output is

$$\mathbf{Z}_l^{fp} = \mathcal{G}_l(\mathbf{A}_l^{fp}). \quad (17)$$

For the quantized block, quantization in the MLP branch and activation perturbations introduce additional errors $\Delta \mathbf{Z}_l^{\text{mlp}}$. Around the full-precision forward path, the quantized block output can be approximated as

$$\mathbf{Z}_l^q \approx \mathcal{G}_l(\mathbf{A}_l^{fp} - \Delta \mathbf{A}_l) - \Delta \mathbf{Z}_l^{\text{mlp}}. \quad (18)$$

Applying a first-order Taylor expansion gives

$$\mathcal{G}_l(\mathbf{A}_l^{fp} - \Delta \mathbf{A}_l) \approx \mathcal{G}_l(\mathbf{A}_l^{fp}) - \mathbf{J}_l \Delta \mathbf{A}_l, \quad (19)$$

where

$$\mathbf{J}_l = \left. \frac{\partial \mathcal{G}_l(\mathbf{A})}{\partial \mathbf{A}} \right|_{\mathbf{A}=\mathbf{A}_l^{fp}} \quad (20)$$

is the local Jacobian of the post-attention transformation. The final block output error is then

$$\Delta \mathbf{Z}_l = \mathbf{Z}_l^{fp} - \mathbf{Z}_l^q \approx \mathbf{J}_l \Delta \mathbf{A}_l + \Delta \mathbf{Z}_l^{\text{mlp}}. \quad (21)$$

Substituting Eq. (16), we obtain

$$\Delta \mathbf{Z}_l \approx \sum_{h=1}^{N_h} \mathbf{J}_l (\Delta \mathbf{H}_{l,h} \mathbf{W}_{o,l,h}) + \Delta \mathbf{Z}_l^{\text{mlp}}. \quad (22)$$

Therefore, the block-level reconstruction loss can be approximated as

$$\mathcal{L}_{\text{rec}} \approx \left\| \sum_{h=1}^{N_h} \mathbf{J}_l (\Delta \mathbf{H}_{l,h} \mathbf{W}_{o,l,h}) + \Delta \mathbf{Z}_l^{\text{mlp}} \right\|_F^2. \quad (23)$$

This expression shows that block-level reconstruction observes a mixed sum of projected head-wise errors. To see that cancellation can hide nonzero head-wise errors, consider a simplified two-head case with $\mathbf{J}_l = \mathbf{I}$, $\mathbf{W}_{o,l,1} = \mathbf{W}_{o,l,2} = \mathbf{\Gamma}$, $\Delta \mathbf{Z}_l^{\text{mlp}} = \mathbf{0}$, $\Delta \mathbf{H}_{l,1} = \mathbf{E}$, and $\Delta \mathbf{H}_{l,2} = -\mathbf{E}$. Then

$$\Delta \mathbf{Z}_l \approx \Delta \mathbf{H}_{l,1} \mathbf{\Gamma} + \Delta \mathbf{H}_{l,2} \mathbf{\Gamma} = \mathbf{0}, \quad (24)$$

which yields

$$\mathcal{L}_{\text{rec}} \approx 0. \quad (25)$$

However, the accumulated head-wise error remains large:

$$\sum_{h=1}^2 \|\Delta \mathbf{H}_{l,h}\|_F^2 = 2 \|\mathbf{E}\|_F^2. \quad (26)$$

Thus, a small block-level reconstruction loss does not necessarily imply small per-head quantization errors. ■

B Variational Interpretation of HEAL

We analyze the PTQ calibration from a variational view to further explain why conventional block-level reconstruction can be insufficient for post-training quantization calibration. We first denote the pre-projection attention-head representations as latent variables by $\mathbf{U}_l$. The full-precision LLM induces a reference distribution $q(\mathbf{U}_l|\mathbf{X}_l)$ over these latent head representations, and the quantized LLM induces a surrogate distribution $p_\theta(\mathbf{U}_l|\mathbf{X}_l)$, where θ_l denotes the calibration-related parameters optimized by the underlying PTQ method for the l -th block. The marginal likelihood of reconstructing the full-precision block output can be written as:

$$\log p_\theta(\mathbf{Z}_l^{fp}|\mathbf{X}_l) = \log \int p_\theta(\mathbf{Z}_l^{fp}, \mathbf{U}_l|\mathbf{X}_l) d\mathbf{U}_l, \quad (27)$$

where $\mathbf{X}_l$ denotes the input hidden states of the l -th Transformer block. By applying the reference distribution $q(\mathbf{U}_l|\mathbf{X}_l)$ and Jensen's inequality, we can obtain the following evidence lower bound:

$$\begin{aligned} \log p_\theta(\mathbf{Z}_l^{fp}|\mathbf{X}_l) &= \log \int q(\mathbf{U}_l|\mathbf{X}_l) \frac{p_\theta(\mathbf{Z}_l^{fp}, \mathbf{U}_l|\mathbf{X}_l)}{q(\mathbf{U}_l|\mathbf{X}_l)} d\mathbf{U}_l \\ &\geq \mathbf{E}_q \left[\log p_\theta(\mathbf{Z}_l^{fp}|\mathbf{U}_l, \mathbf{X}_l) \right] - \text{KL}(q(\mathbf{U}_l|\mathbf{X}_l) \| p_\theta(\mathbf{U}_l|\mathbf{X}_l)). \end{aligned} \quad (28)$$

The first term of the lower bound corresponds to the observation-level reconstruction objective, which encourages the quantized block output to match the full-precision block output. This is consistent with the conventional block-level reconstruction loss used in reconstruction-based PTQ

methods. The second term measures the discrepancy between the full-precision and quantized latent head representations. Therefore, optimizing only the block-level reconstruction objective can be interpreted as a loose variational relaxation that ignores the latent head-alignment term.

To obtain a tractable instantiation of the variational objective, we approximate both the observation distribution and the latent head representation distribution with Gaussian distributions (Ahn et al., 2019; Huang et al., 2024b; Kingma and Welling, 2013). They only provide a convenient approximation under which the corresponding negative log-likelihood and KL divergence admit closed-form objectives. At the observation level, we instantiate the conditional surrogate likelihood of the full-precision block output as a Gaussian centered at the quantized block output:

$$p_\theta(\mathbf{Z}_l^{fp}|\mathbf{U}_l, \mathbf{X}_l) = \mathcal{N}(\mathbf{Z}_l^{fp}; \mathbf{Z}_l^q, \sigma_z^2 \mathbf{I}). \quad (29)$$

Then, the negative log-likelihood term can be written as:

$$-\log p_\theta(\mathbf{Z}_l^{fp}|\mathbf{U}_l, \mathbf{X}_l) = \frac{1}{2\sigma_z^2} \|\mathbf{Z}_l^q - \mathbf{Z}_l^{fp}\|_F^2 + C_z, \quad (30)$$

which corresponds to the conventional block-level reconstruction loss. Similarly, for each attention head, we approximate the full-precision and quantized latent head representations with surrogate Gaussian distributions sharing a fixed isotropic covariance. Their means are parameterized by the corresponding full-precision and quantized head representations, respectively:

$$q(\mathbf{U}_{l,h}|\mathbf{X}_l) = \mathcal{N}(\mathbf{U}_{l,h}; \mathbf{H}_{l,h}^{fp}, \sigma_h^2 \mathbf{I}), \quad (31)$$

$$p_\theta(\mathbf{U}_{l,h}|\mathbf{X}_l) = \mathcal{N}(\mathbf{U}_{l,h}; \mathbf{H}_{l,h}^q, \sigma_h^2 \mathbf{I}). \quad (32)$$

Since these two Gaussian distributions share the same covariance, their KL divergence reduces to the squared distance between their means:

$$\text{KL}(q(\mathbf{U}_{l,h}|\mathbf{X}_l) \| p_\theta(\mathbf{U}_{l,h}|\mathbf{X}_l)) = \frac{\|\mathbf{H}_{l,h}^q - \mathbf{H}_{l,h}^{fp}\|_F^2}{2\sigma_h^2} + C_h. \quad (33)$$

Therefore, for all attention heads, the latent alignment term can be written as:

$$\text{KL}(q(\mathbf{U}_l|\mathbf{X}_l) \| p_\theta(\mathbf{U}_l|\mathbf{X}_l)) = \frac{\sum_{h=1}^{N_h} \|\mathbf{H}_{l,h}^q - \mathbf{H}_{l,h}^{fp}\|_F^2}{2\sigma_h^2} + C. \quad (34)$$

MODEL	METHOD	ARC-C	ARC-E	BOOLQ	HELLASWAG	PIQA	WINOGRANDE	AVG
LLAMA2-7B	QUAROT	36.26	63.43	69.14	65.45	73.12	60.06	61.24
	QUAROT+HEAL	37.20	62.96	71.56	66.53	73.01	61.40	62.11
	SPINQUANT	39.33	66.96	69.42	71.70	74.76	66.06	64.71
	SPINQUANT+HEAL	40.44	68.39	69.66	72.11	76.06	64.80	65.24
LLAMA2-13B	QUAROT	44.62	71.51	74.43	72.36	76.99	65.51	67.57
	QUAROT+HEAL	45.31	72.90	77.00	73.00	77.09	67.40	68.78
	SPINQUANT	45.99	73.40	78.84	74.93	77.45	69.93	70.09
	SPINQUANT+HEAL	46.25	73.57	78.07	75.21	78.56	69.92	70.26
LLAMA3.1-8B	QUAROT	38.91	66.92	72.35	66.59	74.05	64.56	63.90
	QUAROT+HEAL	44.37	71.13	76.51	70.36	76.12	67.48	67.66
	SPINQUANT	43.69	70.41	78.81	73.93	78.24	63.18	68.04
	SPINQUANT+HEAL	47.44	73.36	76.18	73.65	77.31	68.43	69.40

Table 5: The detailed zero-shot results of combining HEAL under W4A4 setting.

Then, minimizing the negative form of the lower bound, we can obtain:

$$\mathcal{L}_{\text{var}} = \frac{\|\mathbf{Z}_l^q - \mathbf{Z}_l^{fp}\|_F^2}{2\sigma_z^2} + \frac{\sum_{h=1}^{N_h} \|\mathbf{H}_{l,h}^q - \mathbf{H}_{l,h}^{fp}\|_F^2}{2\sigma_h^2} + C. \quad (35)$$

Absorbing the constant term C and the variance-dependent coefficients into a balancing hyperparameter λ , we can obtain the final optimization objective:

$$\mathcal{L} = \|\mathbf{Z}_l^q - \mathbf{Z}_l^{fp}\|_F^2 + \lambda \sum_{h=1}^{N_h} \|\mathbf{H}_{l,h}^q - \mathbf{H}_{l,h}^{fp}\|_F^2. \quad (36)$$

C Efficiency Analysis

We evaluate the inference efficiency of different quantization methods and their HEAL-enhanced variants on LLaMA-2-7B. We report the latency of these methods with a sequence length of 2048, a batch size of 64, and 256 decoding steps. As shown in Table 6, HEAL achieves nearly identical latency and memory usage compared with the corresponding QuaRot and FlatQuant baselines, confirming that the proposed calibration objective HEAL improves quantization quality without introducing inference-time overhead and modifying the inference graph of the underlying PTQ method.

METHOD	DECODE SPEEDUP	PREFILL SPEEDUP	MEMORY SAVING FACTOR
QUAROT	$\times 1.79$	$\times 2.14$	$\times 2.15$
QUAROT+HEAL	$\times 1.81$	$\times 2.13$	$\times 2.15$
FLATQUANT	$\times 1.66$	$\times 2.28$	$\times 2.15$
FLATQUANT+HEAL	$\times 1.65$	$\times 2.29$	$\times 2.15$

Table 6: Inference latency and memory usage compared with the FP16 baseline. $\times$ denotes the relative speedup or memory saving factor over the FP16 baseline.

D Analysis of the Head-wise Alignment Loss Scale

We report the median ratio across all calibrated layers at each calibration iteration on Qwen2.5-7B as shown in Figure 7. The raw head-wise alignment loss has a much larger numerical scale than the block-level reconstruction loss, while the weighted contribution remains moderate after applying λ_{head} and the warmup-ramp coefficient α_t . This shows that the small λ_{head} mainly serves to balance the loss scales.

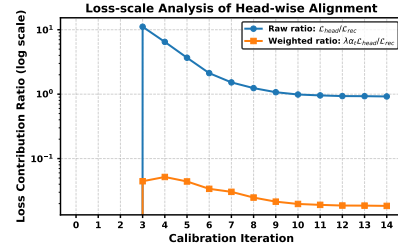


Figure 7: The loss-scale analysis of head alignment.

E Additional Experimental Results

Detailed zero-shot results of combining HEAL.

Table 5 reports the detailed zero-shot performance of combining HEAL with two representative W4A4 quantization baselines, QuaRot and SpinQuant. Overall, HEAL is used to optimize clipping ratios and improves the average accuracy across all evaluated models and baselines. Notably, the improvement is particularly significant on LLaMA3.1-8B, where HEAL enhances QuaRot across all zero-shot tasks. These results demonstrate that HEAL is complementary to existing transformation-based PTQ methods and can further improve the performance of these methods.

Detailed zero-shot results of using GPTQ.

Table 7 reports the detailed zero-shot performance

MODEL	METHOD	ARC-C	ARC-E	BOOLQ	HELLASWAG	PIQA	WINOGRANDE	AVG
LLAMA2-7B	QUAROT	42.32	70.58	72.87	72.62	77.75	66.77	67.15
	SPINQUANT	43.00	71.89	73.67	73.65	76.22	65.51	67.32
	FLATQUANT	44.54	73.19	75.90	74.28	77.42	67.40	68.79
	OURS	43.94	73.78	75.90	74.10	78.02	70.24	69.33
LLAMA2-13B	QUAROT	48.21	74.12	78.35	76.49	78.84	64.26	70.05
	SPINQUANT	48.98	76.01	78.56	77.10	79.27	68.75	71.45
	FLATQUANT	46.84	75.80	78.87	77.71	79.49	70.17	71.48
	OURS	48.04	76.68	79.42	77.80	79.22	71.59	72.13
LLAMA3.1-8B	QUAROT	45.56	74.12	75.17	72.03	75.63	67.38	68.32
	SPINQUANT	46.42	74.28	74.53	74.74	78.18	68.75	69.48
	FLATQUANT	50.26	78.07	80.03	77.00	79.98	72.14	72.91
	OURS	52.13	79.59	80.06	76.46	79.54	72.45	73.37
QWEN2.5-7B	QUAROT	44.03	69.32	76.45	72.19	76.01	65.11	67.19
	SPINQUANT	49.23	77.69	81.53	77.46	77.09	65.82	71.47
	FLATQUANT	49.23	76.81	83.94	76.28	78.56	71.27	72.68
	OURS	52.65	78.58	83.82	76.67	78.29	70.64	73.44
QWEN2.5-14B	QUAROT	51.11	74.71	77.71	77.25	79.54	69.93	71.71
	SPINQUANT	55.20	79.97	80.52	77.93	78.56	69.53	73.62
	FLATQUANT	57.94	80.77	84.16	81.27	81.50	75.16	76.80
	OURS	58.02	81.06	84.13	81.50	80.96	75.53	76.87
QWEN2.5-32B	QUAROT	52.13	79.42	85.57	79.92	79.76	70.80	74.60
	SPINQUANT	51.37	75.42	83.06	80.85	81.12	70.32	73.69
	FLATQUANT	57.68	80.35	87.58	83.14	81.72	74.59	77.51
	OURS	57.94	80.13	87.46	83.29	81.77	75.45	77.67

Table 7: The detailed zero-shot results of using GPTQ quantizer under W4A4 setting. Bold results highlight the best performance.

of different GPTQ-based quantization methods under the W4A4 setting. Compared with QuaRot, SpinQuant and FlatQuant, our method consistently improves the average zero-shot accuracy on all evaluated LLMs. Moreover, our method achieves the best average performance and obtains the best or tied-best results on most zero-shot tasks.

Few-shot results of different methods. Beyond zero-shot evaluation, we also assess the few-shot(5-shot) performance of our method on Qwen2.5-7B. As shown in Table 8, our method consistently outperforms FlatQuant on ReCoRD, SocialIQA, CommonsenseQA, and MMLU. These results demonstrate that our approach generalizes well beyond zero-shot settings and effectively preserves downstream reasoning and knowledge capabilities.

METHOD	RECORD	SOCIALIQA	COMMONSENSEQA	MMLU
FLATQUANT	83.01	55.42	80.84	69.69
OURS	86.30	56.70	83.37	72.03

Table 8: The few-shot results of different methods.

Different surrogate distributions. Our formulation does not assume that attention-head representations truly follow a Gaussian distribution. The

Gaussian surrogate is introduced solely to instantiate the latent alignment term as an optimizable squared Frobenius-norm loss. To further examine whether the effectiveness of HEAL depends on this particular Frobenius-norm-based formulation, we also consider more robust alternative alignment objectives, including an ℓ_1 loss motivated by the Laplace surrogate and a Huber loss based on a Huber-type robust surrogate. We report the results with these different alignment objectives are reported in Table 9.

Compare with another PTQ enhancement method. We compare HEAL with another PTQ enhancement method, GuidedQuant, when both are integrated with SpinQuant under the W4A4 setting in Table 10. HEAL achieves higher average zero-shot accuracy on both LLaMA2-7B and LLaMA2-13B. These results show that HEAL is compatible with existing PTQ pipelines and provides competitive calibration enhancement.

MODEL	SURROGATE DISTRIBUTION	WIKITEXT2	ARC-C	ARC-E	BOOLQ	HELLASWAG	PIQA	WINOGRANDE	AVG
LLAMA3.1-8B	LAPLACE	6.96	50.60	78.70	80.21	76.59	79.11	70.56	72.63
	HUBER-TYPE	6.96	50.43	78.58	80.24	76.71	80.20	72.14	73.05
	GAUSSIAN	6.94	52.13	78.16	80.70	76.97	80.20	72.38	73.42
QWEN2.5-7B	LAPLACE	7.28	51.45	76.94	84.04	77.30	79.54	71.27	73.42
	HUBER-TYPE	7.30	53.50	78.11	84.50	77.24	78.29	70.56	73.70
	GAUSSIAN	7.29	52.22	78.96	83.85	77.38	79.00	70.24	73.61

Table 9: The PPL and zero-shot results of using different surrogate distributions.

MODEL	METHOD	ARC-C	ARC-E	BOOLQ	HELLASWAG	PIQA	WINOGRANDE	AVG
LLAMA2-7B	SPINQUANT+GUIDEDQUANT	39.82	65.34	72.75	71.14	75.79	65.04	64.98
	SPINQUANT+HEAL	40.44	68.39	69.66	72.11	76.06	64.80	65.24
LLAMA2-13B	SPINQUANT+GUIDEDQUANT	46.84	74.75	77.28	74.27	79.05	68.98	70.20
	SPINQUANT+HEAL	46.25	73.57	78.07	75.21	78.56	69.92	70.26

Table 10: The detailed zero-shot results of another PTQ enhancement method under W4A4 setting.