NOT ALL RECONSTRUCTION ERRORS MATTER: SCORE-GUIDED BLOCK-WISE RECONSTRUCTION FOR LOW-BIT LLM QUANTIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Block-wise reconstruction has been widely used for post-training quantization (PTQ) of large language models (LLMs), but conventional objectives penalize all block-output perturbation directions equally despite their different effects on the final predictive distribution. In this paper, we propose SABRE, a score-guided anisotropic reconstruction approach that accounts for these unequal predictive effects during block-wise reconstruction. To identify prediction-sensitive directions, SABRE approximates the Fisher curvature of the full-precision (FP) predictive distribution using rank-one outer products of sampled score vectors. Based on this approximation, SABRE assigns additional penalties to errors aligned with these directions while retaining isotropic reconstruction to control the overall perturbation magnitude. To ensure stable weighting across blocks and tokens, we further introduce gradient-norm normalization that bounds token-dependent directional weights, thereby reducing sensitivity to variations in score-vector scale. The FP-derived score vectors are computed once and cached, enabling efficient block-local reconstruction without repeated end-to-end evaluation. Because SABRE modifies only the reconstruction objective, it can be readily integrated into existing block-wise quantization methods without changing their optimization variables or inference-time computation. Extensive experiments across diverse quantization methods, LLM architectures, and low-bit settings demonstrate consistent perplexity improvements, substantial recovery under aggressive 2-bit quantization, and broad downstream gains.

1 INTRODUCTION

Post-training quantization (PTQ) reduces the memory and computational costs of large language models (LLMs) by representing their parameters at low precision. Under aggressive low-bit quantization, however, limited representational capacity makes it difficult to preserve the behavior of the original full-precision (FP) model. To mitigate the resulting accuracy degradation, block-wise PTQ partitions the model into blocks (e.g., Transformer blocks) and optimizes each quantized block to reconstruct its FP output (Li et al., 2021). This block-wise reconstruction strategy provides a tractable alternative to costly end-to-end optimization and underlies various quantization methods, including FlexRound (Lee et al., 2023), OmniQuant (Shao et al., 2024), and Block-AP (Chen et al., 2025).

Despite its practical effectiveness, conventional block-wise reconstruction does not directly account for how local errors affect the model's final predictions. Most existing objectives minimize the mean squared error (MSE) between FP and quantized block outputs, thereby treating all perturbation directions equally. However, equal-norm block-output perturbations can have substantially different effects on the final predictive distribution (Lee et al., 2026). When exact reconstruction is infeasible, as is often the case under aggressive low-bit quantization, preserving the model's predictive behavior requires considering not only the magnitude of the reconstruction error but also its direction.

Recent studies have incorporated downstream information into local quantization objectives to better preserve the model's predictive behavior (Kim et al., 2025; Tseng et al., 2025; Lee et al., 2026). These methods highlight the importance of accounting for how local quantization errors affect the model's final predictive distribution, which we refer to as *predictive sensitivity*. However, a ques-

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

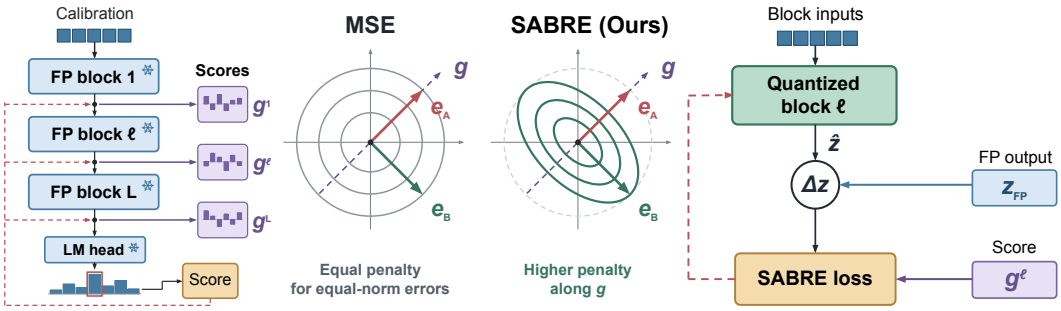


Figure 1: **Overview of SABRE.** **(a)** Position-wise score vectors $\mathbf{g}_t^{(\ell)}$ are computed in a single backward pass from sampled FP token log-probabilities and cached for block-wise quantization. **(b)** Unlike standard MSE, which penalizes equal-norm perturbations identically, SABRE assigns direction-dependent penalties using these score vectors. **(c)** Despite using information from the final predictive distribution, quantization remains block-local: the cached FP output $\mathbf{z}_{\mathrm{FP},t}^{(\ell)}$ and score vector $\mathbf{g}_t^{(\ell)}$ guide the current block without repeated end-to-end evaluation.

tion remains for block-wise PTQ: how can predictive sensitivity be incorporated into a block-local reconstruction objective applicable throughout the model without repeated end-to-end evaluation?

To address this question, we introduce SABRE, a score-guided anisotropic reconstruction approach that accounts for the unequal predictive effects of block-output errors. Our key idea is to use score vectors derived from the FP model to identify prediction-sensitive error directions. Specifically, we combine a directional penalty defined by these score vectors with standard reconstruction, thereby prioritizing errors that more strongly affect the final predictive distribution while controlling the overall error magnitude. We further introduce gradient-norm normalization to bound the token-dependent penalty along prediction-sensitive directions and ensure stable weighting across blocks and tokens.

Our contributions are summarized as follows:

- We introduce a score-guided anisotropic reconstruction approach, SABRE, for block-wise quantization. Our approach uses FP-derived score vectors to prioritize the reduction of errors that affect the final predictive distribution while controlling the overall reconstruction error. These score vectors are computed once and cached to avoid repeated end-to-end evaluation during reconstruction.
- We introduce gradient-norm normalization that bounds the additional weight on prediction-sensitive directions and reduces sensitivity to gradient-scale variations across blocks and tokens. This retains relative predictive sensitivity while preventing excessive directional amplification.
- We demonstrate that SABRE can be readily integrated into existing block-wise quantization methods. Extensive experiments across multiple quantization methods, diverse LLM architectures, and low-bit settings show consistent perplexity improvements, substantial recovery under 2-bit quantization, and broad downstream gains.

2 BACKGROUND

2.1 BLOCK-WISE RECONSTRUCTION-BASED QUANTIZATION

Block-wise quantization partitions a pretrained model into blocks, such as Transformer blocks, and quantizes each block by minimizing the task-loss degradation induced by its output perturbation. Neglecting the first-order term, the quadratic objective for block ℓ is (Li et al., 2021)

$$\min_{\Delta \mathbf{z}^{(\ell)}} \left(\Delta \mathbf{z}^{(\ell)}\right)^T \mathbf{H}^{(\mathbf{z}^{(\ell)})} \Delta \mathbf{z}^{(\ell)}, \tag{1}$$

where $\Delta \mathbf{z}^{(\ell)}$ denotes the block-output perturbation induced by the quantization parameters and $\mathbf{H}^{(\mathbf{z}^{(\ell)})}$ represents the corresponding task-loss curvature. In practice, most methods minimize the

MSE between FP and quantized block outputs, equivalent to approximating the block-output curvature $\mathbf{H}^{(\mathbf{z}^{(\ell)})}$ by the identity matrix:

$$\min_{\Delta\mathbf{z}^{(\ell)}} \|\Delta\mathbf{z}^{(\ell)}\|_2^2. \tag{2}$$

This principle underlies various quantization methods, including FlexRound (Lee et al., 2023), OmniQuant (Shao et al., 2024), and Block-AP (Chen et al., 2025). Compared with layer-wise methods such as GPTQ (Frantar et al., 2022), block-wise methods capture interactions among layers within a block, making them a practical approach to LLM quantization. Appendix A reviews these methods.

2.2 Beyond Isotropic Reconstruction in PTQ

Despite its effectiveness, standard block-wise reconstruction penalizes all perturbation directions equally. This isotropic treatment assigns the same cost to equal-norm block-output perturbations, even when their effects on the final predictive distribution differ substantially (Lee et al., 2026).[1] Preserving the model's predictive behavior therefore motivates objectives that distinguish errors by their downstream consequences rather than their magnitudes alone.

Several methods have incorporated downstream information into local quantization objectives. GuidedQuant applies end-loss gradients to layer-output errors, introducing feature-dependent weighting (Kim et al., 2025). YAQA uses an approximation to the curvature of the final output-distribution discrepancy to guide weight-space adaptive rounding (Tseng et al., 2025). LFQ aligns the FP and quantized predictive distributions by applying a logit-level cross-entropy objective to the final Transformer block (Lee et al., 2026). REAL-Q uses an end-loss-derived Fisher objective, aggregated across tokens, to refine unquantized weights in GPTQ-style quantization (Zhang et al., 2026).

Our method incorporates predictive sensitivity in the block-output space using FP-derived, token-specific score vectors, while stabilizing directional weighting through gradient-norm normalization. Once cached, these score vectors guide the reconstruction of any Transformer block without repeated end-to-end evaluation or projection into the large vocabulary space. The resulting objective preserves cross-feature interactions and can be readily integrated into existing block-wise quantization frameworks without modifying their optimization variables.

2.3 Predictive Curvature via Fisher Information

The second-order formulation in (1) requires a Hessian capturing the effect of block-output perturbations on the task loss. While the conventional isotropic reconstruction objective in (2) is equivalent to approximating this Hessian by the identity matrix, we incorporate predictive sensitivity by considering an output-distribution surrogate. Specifically, rather than directly approximating the task-loss Hessian, we use the KL divergence between the fixed FP predictive distribution and that induced by a block-output perturbation as a surrogate for task-loss degradation. At the unperturbed FP reference, the Hessian of this KL divergence coincides with the Fisher information matrix (Tseng et al., 2025) and can be expressed in terms of score vectors (gradients of output-token log-probabilities):

$$\mathbf{F}^{(\mathbf{z}^{(\ell)})} = \mathbb{E}_{v\sim p_{\mathrm{FP}}}\left[\nabla_{\mathbf{z}^{(\ell)}} \log p_{\mathrm{FP}}(v)\,(\nabla_{\mathbf{z}^{(\ell)}} \log p_{\mathrm{FP}}(v))^T\right], \tag{3}$$

where $p_{\mathrm{FP}}(v) = \prod_{s=1}^{T} p_{\mathrm{FP},s}(v_s)$ is the product distribution obtained by independently sampling $v_s \sim p_{\mathrm{FP},s}$ at each position of a fixed calibration sequence, and the score vectors are evaluated at the FP block outputs. This formulation shows that the FP-derived score vectors encode directional information about predictive sensitivity in the block-output space. In this work, we use these score vectors to construct an anisotropic block-wise reconstruction objective.

3 Method

3.1 SABRE: Score-Guided Anisotropic Block-wise Reconstruction

We construct an anisotropic reconstruction objective that assigns different penalties to block-output errors based on their alignment with FP-derived score directions. Explicitly evaluating the expecta-

[1] We demonstrate this directional disparity through a controlled perturbation experiment in Appendix B.

tion in (3) is computationally expensive, as it requires aggregating score contributions over the entire vocabulary. To avoid this cost, we instead use a Monte Carlo estimator of this expectation. Specifically, for each predictive position s, we sample $v_s \sim p_{\mathrm{FP},s}$ from the FP predictive distribution at that position. Then, for each block-output position t, we compute the downstream score vector

$$\mathbf{g}_t^{(\ell)} = \nabla_{\mathbf{z}_t^{(\ell)}} \sum_{s=t}^{T} \log p_{\mathrm{FP},s}(v_s), \tag{4}$$

where $\mathbf{z}_t^{(\ell)}$ denotes the output of the ℓ-th block at position t. By causality, $\mathbf{z}_t^{(\ell)}$ can affect only predictions at positions $s \geq t$.[2] The resulting outer product provides a stochastic rank-one approximation[3] to the Fisher matrix associated with $\mathbf{z}_t^{(\ell)}$:

$$\mathbf{F}^{(\mathbf{z}_t^{(\ell)})} \approx \mathbf{g}_t^{(\ell)} \left(\mathbf{g}_t^{(\ell)}\right)^T. \tag{5}$$

Ignoring cross-token terms, we sum the token-wise penalties to obtain the score-guided objective:

$$\mathcal{L}_{\mathrm{SG}} = \sum_t \left(\Delta\mathbf{z}_t^{(\ell)}\right)^T \mathbf{g}_t^{(\ell)} \left(\mathbf{g}_t^{(\ell)}\right)^T \Delta\mathbf{z}_t^{(\ell)} = \sum_t \langle \mathbf{g}_t^{(\ell)}, \Delta\mathbf{z}_t^{(\ell)} \rangle^2. \tag{6}$$

However, minimizing (6) alone does not control the overall block-output perturbation, since the objective only penalizes the component of $\Delta\mathbf{z}_t^{(\ell)}$ along the sampled score direction $\mathbf{g}_t^{(\ell)}$. Specifically, any perturbation orthogonal to $\mathbf{g}_t^{(\ell)}$ does not affect this directional loss, even if its magnitude is large. Indeed, we observe that the directional term alone leads to severe degradation without magnitude control (see Section 4.3).

We therefore augment the standard reconstruction loss with the score-guided directional penalty:

$$\mathcal{L}_{\text{SG-reg}} = \sum_t \left(\left\| \Delta\mathbf{z}_t^{(\ell)} \right\|_2^2 + \lambda_t^{(\ell)} \langle \mathbf{g}_t^{(\ell)}, \Delta\mathbf{z}_t^{(\ell)} \rangle^2 \right), \tag{7}$$

where $\lambda_t^{(\ell)} > 0$ is a token-dependent coefficient that controls the score-guided directional penalty. To clarify the effect of this regularization, consider a nonzero score vector $\mathbf{g}_t^{(\ell)}$ and decompose $\Delta\mathbf{z}_t^{(\ell)}$ into components parallel and orthogonal to its direction:

$$\Delta\mathbf{z}_t^{(\ell)} = \mathbf{P}_{\parallel} \Delta\mathbf{z}_t^{(\ell)} + \mathbf{P}_{\perp} \Delta\mathbf{z}_t^{(\ell)}, \tag{8}$$

where $\mathbf{P}_{\parallel} \Delta\mathbf{z}_t^{(\ell)}$ denotes the component parallel to $\mathbf{g}_t^{(\ell)}$, and $\mathbf{P}_{\perp} \Delta\mathbf{z}_t^{(\ell)}$ denotes the orthogonal component. The loss for each token position can be rewritten as (see Appendix C for details)

$$\left\| \Delta\mathbf{z}_t^{(\ell)} \right\|_2^2 + \lambda_t^{(\ell)} \langle \mathbf{g}_t^{(\ell)}, \Delta\mathbf{z}_t^{(\ell)} \rangle^2 = \left\| \mathbf{P}_{\perp} \Delta\mathbf{z}_t^{(\ell)} \right\|_2^2 + \left(1 + \lambda_t^{(\ell)} \left\| \mathbf{g}_t^{(\ell)} \right\|_2^2\right) \left\| \mathbf{P}_{\parallel} \Delta\mathbf{z}_t^{(\ell)} \right\|_2^2. \tag{9}$$

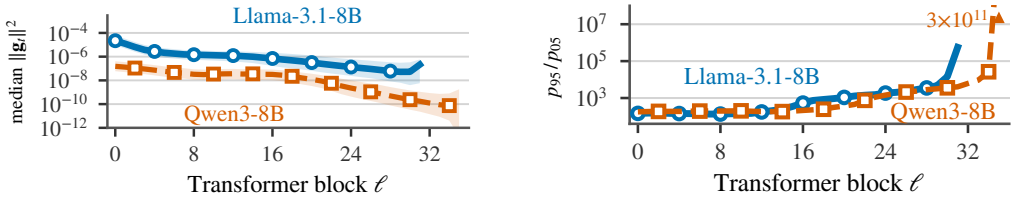
This decomposition shows that the standard reconstruction term continues to control perturbations in all directions, while the score-guided term places additional weight on perturbations parallel to the sampled score direction. Moreover, the score direction and the strength of this additional weighting vary across token positions, making the proposed objective both token-conditioned and anisotropic.

3.2 Gradient-Normalized Directional Weighting

As shown in (9), the weight on the score-parallel component is determined by $1 + \lambda_t^{(\ell)} \|\mathbf{g}_t^{(\ell)}\|_2^2$. If $\lambda_t^{(\ell)}$ is fixed across tokens, this weight can vary substantially with the gradient norm, as confirmed by the considerable variation shown in Fig. 2. Consequently, unnormalized directional weighting can be highly sensitive to gradient-scale variations across blocks and tokens.

[2] For the final Transformer block, subsequent operations are position-wise, so only $s = t$ contributes.

[3] Sampling M labels per token generalizes this estimator to a rank-at-most-M Monte Carlo approximation. However, as shown in Appendix F, additional samples provide limited downstream gains relative to their additional memory and computational costs. We therefore use $M = 1$ throughout our main experiments.



(a) Block-wise median of $\|\mathbf{g}_t^{(\ell)}\|_2^2$ (b) Within-block spread of $\|\mathbf{g}_t^{(\ell)}\|_2^2$ (p_{95}/p_{05})

Figure 2: Variation in squared gradient norms $\|\mathbf{g}_t^{(\ell)}\|_2^2$ across Transformer blocks and calibration tokens. (a) The block-wise median varies substantially with depth, with the shaded regions indicating the interquartile range. (b) The large p_{95}/p_{05} ratios indicate within-block variation across tokens.

To reduce this dependence, we define the token-dependent coefficient as

$$\lambda_t^{(\ell)} = \frac{\lambda^{(\ell)}}{\tau^{(\ell)} + \|\mathbf{g}_t^{(\ell)}\|_2^2}, \tag{10}$$

where $\lambda^{(\ell)} > 0$ controls the directional strength for block ℓ, and $\tau^{(\ell)} > 0$ is a block-specific normalization scale. With this choice, the weight on the score-parallel component becomes

$$1 + \lambda_t^{(\ell)} \|\mathbf{g}_t^{(\ell)}\|_2^2 = 1 + \lambda^{(\ell)} \frac{\|\mathbf{g}_t^{(\ell)}\|_2^2}{\tau^{(\ell)} + \|\mathbf{g}_t^{(\ell)}\|_2^2}, \tag{11}$$

which lies in $[1, 1 + \lambda^{(\ell)})$, thereby bounding the additional directional weight independently of the absolute gradient scale. Substituting (10) into (7) gives the final objective for SABRE:

$$\mathcal{L}_{\text{SABRE}} = \sum_t \left(\|\Delta \mathbf{z}_t^{(\ell)}\|_2^2 + \lambda^{(\ell)} \frac{\langle \mathbf{g}_t^{(\ell)}, \Delta \mathbf{z}_t^{(\ell)} \rangle^2}{\tau^{(\ell)} + \|\mathbf{g}_t^{(\ell)}\|_2^2} \right). \tag{12}$$

Normalization scale. The normalization scale $\tau^{(\ell)}$ does more than ensure numerical stability. Without $\tau^{(\ell)}$, all nonzero-gradient tokens would receive equal additional weight on the score-parallel component, regardless of gradient magnitude. Introducing $\tau^{(\ell)}$ preserves relative magnitude information: the additional weight increases with $\|\mathbf{g}_t^{(\ell)}\|_2^2/\tau^{(\ell)}$ and vanishes as this ratio converges to zero. We set $\tau^{(\ell)}$ to the median nonzero squared gradient norm over tokens in block ℓ and keep it fixed during reconstruction. The median provides an outlier-resistant reference scale and, by construction, maps the median squared gradient norm to an additional score-parallel weight of $\lambda^{(\ell)}/2$.

Depth-dependent weighting. We employ an empirical late-weighted schedule for $\lambda^{(\ell)}$ that progressively increases the directional strength in later blocks:

$$\lambda^{(\ell)} = \lambda \rho^{L-1-\ell}, \qquad \ell = 0, \ldots, L-1, \quad 0 < \rho \leq 1, \tag{13}$$

where L is the number of Transformer blocks, $\lambda > 0$ determines the directional strength at the final block, and ρ controls its variation across depth. This schedule assigns greater directional weight to later blocks, whose outputs are more directly coupled to the final predictive distribution.

Quantization workflow. The FP-derived score vectors are computed once before quantization, cached separately for each block, and kept fixed throughout optimization. We quantize the blocks sequentially using the outputs of previously quantized blocks as inputs, FP block outputs as reconstruction targets, and the corresponding cached score vectors as directional guidance. Each update backpropagates only through the current block, avoiding repeated end-to-end evaluation. Because SABRE modifies only the reconstruction objective, it adds no inference-time overhead.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate SABRE by integrating it into three representative block-wise quantization methods: OmniQuant (Shao et al., 2024), Block-AP (Chen et al., 2025), and FlexRound (Lee et al., 2023). We

Table 1: **Integration with existing block-wise quantization methods under W2g128.** Within each quantization method, the better result is shown in **bold**.

Method	PPL↓	CSR6↑	MMLU↑	IFEval↑	Method	PPL↓	CSR6↑	MMLU↑	IFEval↑
Qwen3-4B-Instruct					**Llama-3.1-8B-Instruct**				
Full precision	9.40	73.0	72.6	82.6	Full precision	6.75	75.9	68.2	73.8
Block-AP	40.82	57.4	48.9	40.3	Block-AP	352.87	48.3	24.7	10.2
+ SABRE	**29.30**	**59.6**	**50.9**	**43.3**	+ SABRE	**22.36**	**61.3**	**42.5**	**26.1**
FlexRound	29.58	62.9	55.1	52.7	FlexRound	1832.63	50.6	48.8	11.3
+ SABRE	**24.71**	**63.3**	**56.5**	**54.9**	+ SABRE	**472.92**	**51.3**	**49.0**	**46.8**
Qwen3-8B					**Qwen2.5-14B-Instruct**				
Full precision	9.00	74.2	74.8	81.0	Full precision	5.25	78.9	79.9	78.2
Block-AP	4.5×10^6	51.4	46.0	21.1	Block-AP	4.5×10^6	63.5	24.3	11.5
+ SABRE	**26.05**	**64.1**	**50.3**	**28.5**	+ SABRE	**13.07**	**69.5**	**57.3**	**47.7**
FlexRound	90.22	65.4	55.2	44.4	FlexRound	2.1×10^7	53.4	26.9	11.6
+ SABRE	**23.44**	**65.8**	**56.6**	**45.5**	+ SABRE	**12.43**	**68.6**	**57.8**	**41.0**

Table 2: **Integration across 3-bit quantization granularities (W3 and W3g128).** For each base method, the better result between standard MSE reconstruction and SABRE is shown in **bold**.

	W3 (per-channel)						W3g128					
Method	PPL↓	CSR6	MMLU	IFEval	GSM8K	MATH500	PPL↓	CSR6	MMLU	IFEval	GSM8K	MATH500
Qwen3-4B-Instruct												
Full precision (FP16)	9.40	73.0	72.6	82.6	92.5	73.8	9.40	73.0	72.6	82.6	92.5	73.8
Block-AP	15.61	68.7	65.2	65.8	77.3	48.2	13.32	69.6	66.3	70.1	79.6	60.2
+ SABRE	**14.96**	**69.1**	**66.2**	**69.3**	**78.8**	**50.4**	**12.43**	**70.0**	**66.7**	**73.9**	**82.2**	**61.0**
OmniQuant	18.77	65.4	58.4	62.8	58.4	36.8	13.22	66.7	64.3	72.5	77.9	59.8
+ SABRE	**14.12**	**67.9**	**61.8**	**63.4**	**62.8**	**39.2**	**11.48**	**68.2**	**65.3**	**72.8**	**80.4**	**60.0**
FlexRound	13.78	**70.5**	68.0	74.1	83.3	66.8	11.52	70.6	68.8	75.2	87.0	68.6
+ SABRE	**13.15**	69.8	**68.3**	**74.5**	**83.9**	**67.4**	**11.07**	**71.1**	**69.3**	**76.7**	87.0	**69.4**
Qwen3-8B												
Full precision (FP16)	9.00	74.2	74.8	81.0	88.3	75.8	9.00	74.2	74.8	81.0	88.3	75.8
Block-AP	12.98	72.5	69.0	67.7	**75.1**	53.8	11.24	**72.9**	70.0	73.6	**79.0**	**62.0**
+ SABRE	**12.43**	**73.2**	**69.7**	**68.2**	74.7	**56.6**	**11.11**	72.6	**70.6**	**75.8**	77.5	61.2
OmniQuant	15.23	67.4	64.3	60.4	42.8	38.0	11.83	71.2	**69.2**	68.9	75.7	56.8
+ SABRE	**11.69**	**69.0**	**64.8**	**67.7**	**58.0**	**42.2**	**10.49**	**71.8**	68.8	**74.7**	**80.4**	**58.4**
FlexRound	11.96	73.8	**71.5**	72.5	81.2	62.4	10.63	**73.8**	71.5	77.3	81.3	67.0
+ SABRE	**11.76**	**73.9**	71.2	**76.5**	**81.7**	**63.8**	**10.24**	73.6	**72.2**	**78.6**	**84.8**	**68.2**

conduct experiments on four models: Qwen3-4B-Instruct, Qwen3-8B (Yang et al., 2025), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and Qwen2.5-14B-Instruct (Qwen et al., 2025). For calibration, we use a mixture of Alpaca (Taori et al., 2023) and C4 (Raffel et al., 2020). We additionally evaluate robustness to the calibration-data composition using C4-only calibration in Appendix H.5.

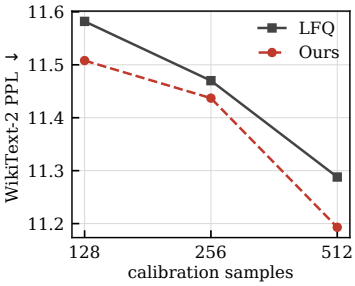
We evaluate language modeling quality using perplexity (PPL) on WikiText2 (Merity et al., 2016) and language understanding using five-shot accuracy on MMLU (Hendrycks et al., 2020). We assess instruction-following and generation using IFEval (Zhou et al., 2023) (prompt-level strict accuracy), GSM8K (Cobbe et al., 2021), and MATH500 (Lightman et al., 2024). For common sense reasoning, we measure zero-shot accuracy on ARC-Challenge, ARC-Easy (Clark et al., 2018), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), OpenBookQA (Mihaylov et al., 2018), and WinoGrande (Sakaguchi et al., 2021), and report the average score as CSR6. Additional experimental details, including the selected hyperparameter values, are provided in Appendix D.

4.2 Integration with Existing Block-wise Quantization Methods

We examine whether SABRE enhances different block-wise quantization methods. Since it modifies only the reconstruction objective, it preserves each method's quantization format and optimization variables, and we evaluate its effectiveness across diverse architectures, bit-widths, and granularities.

Bits	Method	Blocks	PPL↓	CSR6	MMLU	IFEval	Time
W3	MSE	all	13.15	70.2	67.6	74.3	17.4k
	LFQ	last	12.67	70.8	67.6	72.8	28.2k
	SABRE	last	12.47	70.7	67.6	74.7	17.6k
	SABRE	**all**	**12.04**	**71.0**	**68.2**	**76.0**	19.5k
W2g128	MSE	all	29.66	63.3	55.8	53.6	17.5k
	LFQ	last	23.66	64.0	55.6	55.1	28.4k
	SABRE	last	23.55	64.3	55.7	53.2	17.7k
	SABRE	**all**	**22.89**	**64.4**	**55.9**	**58.8**	19.6k

(a) Comparison with LFQ.



(b) Calibration budget.

Figure 3: Comparison with last-block logit distillation (LFQ) on Qwen3-4B-Instruct using Block-AP. (a) LFQ and last-block SABRE optimize only the final block from the same MSE-reconstructed prefix, whereas all-block SABRE optimizes every block. **Bold** marks the best quantized variant. Time (sec) includes prefix reconstruction and one-time score caching on one NVIDIA A100. (b) SABRE achieves lower PPL than LFQ across all calibration budgets under W3g128.

Recovery under W2g128 quantization. As shown in Table 1, SABRE provides substantial improvements in challenging W2g128 settings, particularly when the MSE baseline is severely degraded. Across all evaluated model-method pairs, SABRE consistently improves PPL, CSR6, MMLU, and IFEval. For example, with Block-AP on Llama-3.1-8B-Instruct, SABRE reduces PPL from 352.87 to 22.36 and improves CSR6 and MMLU from 48.3% and 24.7% to 61.3% and 42.5%, respectively, while with FlexRound, IFEval rises from 11.3% to 46.8%. On Qwen2.5-14B-Instruct, both MMLU and IFEval improve by approximately 30 percentage points. These results suggest that selective error reduction along prediction-sensitive directions is particularly beneficial under aggressive quantization, where limited representational capacity makes accurate reconstruction in all directions difficult. Further evidence is provided in Appendix H.2, where SABRE reduces the KL divergence from the FP distribution from 0.946 to 0.837 on Qwen3-4B-Instruct. Moreover, the optimization-seed analysis in Appendix H.4 shows that improvements remain stable across runs.

Transfer across 3-bit configurations. We further evaluate SABRE under both per-channel W3 and group-wise W3g128 quantization on Qwen3-4B-Instruct and Qwen3-8B, with Llama-3.1-8B-Instruct results provided in Appendix H.1. Across all twelve model-method-granularity combinations in Table 2, SABRE consistently reduces PPL, demonstrating robust language-modeling improvements. Benefits generally extend to downstream tasks. For example, with Block-AP on Qwen3-4B-Instruct under W3g128, SABRE improves IFEval from 70.1% to 73.9% and GSM8K from 79.6% to 82.2%. With OmniQuant on Qwen3-8B under W3, SABRE improves IFEval and GSM8K by 7.3 and 15.2 percentage points, respectively. Overall, SABRE serves as an effective objective-level enhancement to existing block-wise quantization methods. For broader context, we compare the resulting integrations against representative training-free methods (e.g., GPTQ and GuidedQuant) in Appendix G and provide additional results on reasoning models in Appendix H.6.

Comparison with LFQ. We compare SABRE with LFQ under a controlled last-block reconstruction setting, where both methods start from the same MSE-reconstructed prefix and optimize only the final Transformer block. As shown in Fig. 3(a), last-block SABRE is approximately $1.5\times$ faster than LFQ, because LFQ repeatedly applies the final normalization and vocabulary projection during optimization while SABRE reuses cached score vectors without these operations. Despite its lower computational cost, last-block SABRE achieves lower PPL than LFQ across all evaluated calibration budgets, as shown in Fig. 3(b). Applying SABRE to every block yields further improvements, demonstrating the benefit of using predictive sensitivity throughout the model.

4.3 Ablation Studies

4.3.1 Component Ablations

We conduct component ablations of SABRE on Qwen3-4B-Instruct under symmetric W3g128 and W2g128 quantization, as summarized in Table 3. Using Block-AP as the base quantization method, we construct each variant by modifying its standard MSE reconstruction objective. All variants op-

Table 3: **Component ablation of SABRE.** Results on Qwen3-4B-Instruct with Block-AP under W3g128 and W2g128 ($\lambda = 1000$, $\rho = 0.94$). **Bold** indicates the better result between MSE and the full SABRE objective. All variants should be compared against both reference objectives.

	W3g128				W2g128			
Loss	PPL↓	CSR6	MMLU	IFEval	PPL↓	CSR6	MMLU	IFEval
MSE (Block-AP)	13.32	69.6	66.3	70.1	40.82	57.4	48.9	40.3
SABRE (full)	**12.43**	**70.0**	**66.7**	**73.9**	**29.30**	**59.6**	**50.9**	**43.3**
Is the magnitude regularizer necessary?								
direction term only (normalized, no MSE)	17.48	63.5	58.2	56.0	6.4×10^5	40.5	24.2	11.3
Is the gradient-norm normalization necessary?								
unnormalized, original strength $\lambda_t{=}\lambda$	12.70	69.6	66.3	71.5	52.31	58.1	50.2	43.1
block-wise scaling only $\lambda_t = \lambda/\tau$	15.02	66.2	61.5	64.3	44.10	54.9	28.2	22.6
Is token-wise conditioning necessary?								
token-shared normalized Fisher (cf. Tseng et al. (2025))	12.87	69.8	66.3	69.7	49.27	59.6	48.9	39.9
shared mean score vector (mini-batch)	13.35	69.4	66.0	72.1	52.97	58.0	50.3	43.1
Is token-score correspondence important?								
token-shuffled score vectors	13.24	69.1	66.2	70.1	45.96	58.8	49.2	38.6
Are cross-feature interactions important?								
coordinate-diagonal approximation $\mathrm{diag}(\mathbf{g}_t^2)$ (cf. Kim et al. (2023))	13.03	69.1	66.1	68.9	40.74	58.1	48.3	42.3

Notes. Rows marked 'cf.' adopt only the curvature structure from the cited method within our block-wise reconstruction objective; they are controlled ablations rather than re-implementations.

timize every Transformer block under the same configuration and differ only in their reconstruction objectives. Further details on the ablation variants are provided in Appendix E.

Magnitude regularization. To assess magnitude regularization, we remove only the MSE term from (12), retaining the normalized directional penalty. The directional-only variant degrades substantially under W3g128 and nearly collapses under W2g128, where PPL increases to 6.4×10^5 and MMLU and IFEval fall to 24.2% and 11.3%, respectively. This occurs because the directional loss cannot control perturbations orthogonal to the sampled score directions. Adding magnitude regularization reduces W2g128 PPL to 29.30 and improves MMLU and IFEval to 50.9% and 43.3%, respectively, outperforming MSE across all three metrics.

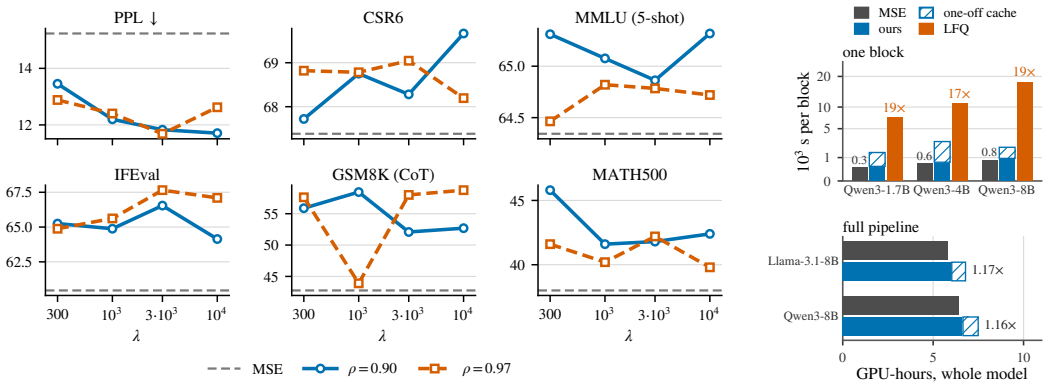
Gradient-norm normalization. We assess the role of gradient-norm normalization using two variants without token-wise normalization. First, setting $\lambda_t^{(\ell)} = \lambda^{(\ell)}$ makes the directional weight proportional to the raw squared gradient norm, as indicated in (9). Although competitive under W3g128, this variant increases W2g128 PPL from 29.30 to 52.31. Second, block-wise scaling with $\lambda_t^{(\ell)} = \lambda^{(\ell)}/\tau^{(\ell)}$ yields PPL of 15.02 and 44.10 under W3g128 and W2g128.[4] These results support the proposed token-wise gradient-norm normalization, which bounds large-gradient contributions and improves W2g128 reconstruction and downstream performance.

Token-wise conditioning. We replace token-specific score information with two shared constructions. The token-shared Fisher averages normalized score outer products over calibration tokens, while the mean-score variant averages score vectors within each mini-batch. Although both variants remain relatively competitive under W3g128, their PPL increases from 29.30 to 49.27 and 52.97, respectively, under W2g128. These shared constructions therefore fail to capture heterogeneous predictive sensitivity across token positions, supporting the use of token-specific score information.

Token-score correspondence. We randomly shuffle score directions across token positions while preserving each token's gradient norm and directional coefficient, disrupting the correspondence between score directions and reconstruction errors. Under W2g128, this variant increases PPL from 29.30 to 45.96 and reduces IFEval by approximately 5 percentage points. Thus, diverse token-specific score vectors alone are insufficient; each error should be paired with its own score direction.

Cross-feature interactions. We replace the score outer product $\mathbf{g}_t^{(\ell)}(\mathbf{g}_t^{(\ell)})^T$ with its coordinate-diagonal approximation, which preserves feature-wise weights but removes all cross-feature interactions. Under W2g128, this approximation increases PPL from 29.30 to 40.74. This demonstrates

[4] Additional strength and KL comparisons appear in Appendix H.3.



(a) Hyperparameter sensitivity on Qwen3-8B with OmniQuant under W3. (b) Training cost on one A100.

Figure 4: (a) Every plotted (λ, ρ) pair improves PPL and all five downstream metrics over MSE (dashed lines). (b) For the final block under W2 per-channel (top), LFQ is 17–19× slower than SABRE; score caching (hatched) accounts for approximately 11% of full-pipeline runtime (bottom).

that independently weighting output features cannot fully represent prediction-sensitive directions and that off-diagonal interactions provide useful directional information.

4.3.2 HYPERPARAMETER SENSITIVITY

We analyze the sensitivity of SABRE to the final-block directional strength λ and the depth-decay factor ρ (see (13)), using Qwen3-8B with OmniQuant under W3 per-channel quantization. As shown in Fig. 4(a), every plotted (λ, ρ) pair improves PPL and all five downstream metrics over the MSE baseline. Although the optimal configuration varies across individual benchmarks, SABRE remains effective over a broad range of λ and ρ, demonstrating robustness to hyperparameter selection.

4.3.3 QUANTIZATION RUNTIME

We compare the quantization cost of SABRE with standard MSE reconstruction on a single NVIDIA A100 GPU. As shown in Fig. 4(b), block-wise reconstruction with SABRE requires only 1.03–1.04× the runtime of MSE reconstruction. Including the one-time score-caching stage, which accounts for approximately 11% of the total runtime, the complete SABRE pipeline requires only 1.16–1.17× the runtime of MSE reconstruction. These results demonstrate that FP-derived predictive information can be incorporated with modest computational overhead. We emphasize that, during final-block reconstruction, LFQ is 17–19× slower per reconstructed block than SABRE because it repeatedly evaluates the final normalization and LM head to compute its output-distribution objective. This per-block ratio is distinct from the full-pipeline comparison above.

5 CONCLUSION

We presented SABRE, a score-guided anisotropic reconstruction approach for low-bit LLM quantization. Using FP-derived score vectors, SABRE emphasizes prediction-sensitive error directions while controlling the overall reconstruction error. Gradient-norm normalization further bounds the directional weighting, and the resulting objective can be readily integrated into existing block-wise quantization methods without additional inference-time computation. Extensive experiments demonstrate the effectiveness of SABRE across diverse quantization methods, LLM architectures, and low-bit settings. These results demonstrate that not all reconstruction errors matter equally and highlight predictive sensitivity to error directions as an effective principle for low-bit LLM quantization.

AI USE STATEMENT

We used generative AI tools to assist with language editing, manuscript organization, and the presentation of figures and tables. All AI-assisted content was reviewed and revised by the authors, and all technical claims, experimental results, and implementations were independently verified. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This work uses publicly available models and datasets and does not involve human subjects or the intentional collection of personally identifiable information. The proposed method is intended to improve the computational efficiency of large language models. We are not aware of any ethical concerns specific to this work beyond those generally associated with the deployment of language models.

REPRODUCIBILITY STATEMENT

Detailed experimental settings, implementation details, hyperparameter configurations, and ablation protocols are provided in the appendices. We also report results across multiple models, quantization methods, calibration datasets, and optimization seeds.

REFERENCES

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv:2503.04697*, 2025.

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

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. EfficientQAT: Efficient quantization-aware training for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10081–10100, 2025.

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

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

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

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv:2407.21783*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv:2501.12948*, 2025.

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

Jinuk Kim, Marwa El Halabi, Wonpyo Park, Clemens JS Schaefer, Deokjae Lee, Yeonhong Park, Jae W Lee, and Hyun Oh Song. GuidedQuant: Large language model quantization via exploiting end loss guidance. *arXiv:2505.07004*, 2025.

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

Jung Hyun Lee, Jeonghoon Kim, Se Jung Kwon, and Dongsoo Lee. FlexRound: Learnable rounding based on element-wise division for post-training quantization. In *International Conference on Machine Learning*, pp. 18913–18939. PMLR, 2023.

Jung Hyun Lee, June Yong Yang, Jungwook Choi, and Eunho Yang. LFQ: Logit-aware final-block quantization for boosting the generation quality of low-bit quantized LLMs. *arXiv:2605.29756*, 2026.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: Pushing the limit of post-training quantization by block reconstruction. *arXiv:2102.05426*, 2021.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2381–2391, 2018.

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, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

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.

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

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

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford Alpaca: An instruction-following LLaMA model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

Albert Tseng, Zhaofeng Sun, and Christopher De Sa. Model-preserving adaptive rounding. *arXiv:2505.22988*, 2025.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Qian Zhang, Yaoming Li, Zhewen Tan, Yanshu Wang, Heng Lu, Kun Su, Zongwei Lv, Wenhan Yu, Yongge Ma, Yinjun Han, Ruikang Liu, and Tong Yang. REAL-Q: E2E LLM quantization via dynamic gradient descent. *arXiv:2609.00049*, 2026.

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

A Block-wise Reconstruction-based Quantization Methods

As discussed in Section 2.1, block-wise quantization methods optimize each block to reproduce its full-precision (FP) output. For weight-only quantization, this process can be formulated as

$$\min_{\phi} \left\| \mathcal{F}^{(\ell)}(\mathcal{Q}(\mathbf{W}^{(\ell)};\phi), \mathbf{X}^{(\ell)}) - \mathcal{F}^{(\ell)}(\mathbf{W}^{(\ell)}, \mathbf{X}^{(\ell)}) \right\|_F^2, \tag{14}$$

where $\mathbf{X}^{(\ell)}$ and $\mathbf{W}^{(\ell)}$ denote the input and FP weights of the ℓ-th block $\mathcal{F}^{(\ell)}$, respectively. Here, the function $\mathcal{Q}(\mathbf{W}^{(\ell)};\phi)$ produces quantized weights according to the optimization variables ϕ, which may include quantization scales, zero-points, clipping parameters, rounding variables, or the block weights themselves. Although FlexRound, OmniQuant, and Block-AP all employ block-wise reconstruction, they differ primarily in their parameterization of ϕ and the corresponding optimization procedure, as summarized below. For brevity, clamping to the representable integer range is omitted from the quantizer definitions, and all operations are broadcast over the corresponding input-channel dimensions. All three methods employ the straight-through estimator to propagate gradients through the non-differentiable rounding operation.

FlexRound introduces a learnable rounding scheme based on element-wise division (Lee et al., 2023). Instead of adjusting rounding decisions through additive offsets, FlexRound divides each pretrained weight by a learnable element-wise factor before rounding. Specifically, for FP weights $\mathbf{W} \in \mathbb{R}^{c_{out} \times c_{in}}$, the FlexRound quantizer is defined as

$$\mathcal{Q}_{\text{FlexRound}}(\mathbf{W}; \mathbf{s}_1, \mathbf{S}_2, \mathbf{s}_3) = \mathbf{s}_1 \left\lfloor \frac{\mathbf{W}}{\mathbf{s}_1 \odot \mathbf{S}_2 \odot \mathbf{s}_3} \right\rceil, \tag{15}$$

where $\mathbf{s}_1 \in \mathbb{R}_{>0}^{c_{out} \times \frac{c_{in}}{g}}$ contains group-wise quantization scales, $\mathbf{S}_2 \in \mathbb{R}_{>0}^{c_{out} \times c_{in}}$ contains element-wise division factors, and $\mathbf{s}_3 \in \mathbb{R}_{>0}^{c_{out} \times 1}$ accounts for variations across output channels. Here, g denotes the group size (number of input-channel elements in each quantization group), with $g = c_{in}$ corresponding to per-channel quantization. During block-wise reconstruction, FlexRound keeps $\mathbf{W}$ fixed and optimizes $\phi_{\text{FlexRound}} = \{\mathbf{s}_1, \mathbf{S}_2, \mathbf{s}_3\}$.

OmniQuant introduces learnable weight clipping (LWC),[5] which optimizes the effective weight range used for quantization (Shao et al., 2024). Specifically, the OmniQuant quantizer is defined as

$$\mathcal{Q}_{\text{OmniQuant}}(\mathbf{W}; \boldsymbol{\beta}, \boldsymbol{\gamma}) = \mathbf{h} \left\lfloor \frac{\mathbf{W}}{\mathbf{h}} \right\rceil, \qquad \mathbf{h} = \frac{\boldsymbol{\gamma} \max(\mathbf{W}) - \boldsymbol{\beta} \min(\mathbf{W})}{2^b - 1}, \tag{16}$$

where b denotes the target bit-width, and the minimum and maximum are computed within each quantization group. During block-wise reconstruction, OmniQuant keeps $\mathbf{W}$ fixed and optimizes $\phi_{\text{OmniQuant}} = \{\boldsymbol{\beta}, \boldsymbol{\gamma}\}$, where $\boldsymbol{\beta}, \boldsymbol{\gamma} \in [0,1]^{c_{out} \times \frac{c_{in}}{g}}$ determine the lower and upper bounds of the effective quantization range, respectively. By adapting the clipping range to the block-wise reconstruction objective, OmniQuant reduces the influence of extreme weight values on the quantization scale.

Block-AP performs block-wise quantization-aware training by jointly optimizing the model weights and quantization scales (Chen et al., 2025). Specifically, the Block-AP quantizer is defined as

$$\mathcal{Q}_{\text{Block-AP}}(\widetilde{\mathbf{W}}; \mathbf{s}) = \mathbf{s} \left\lfloor \frac{\widetilde{\mathbf{W}}}{\mathbf{s}} \right\rceil, \qquad \widetilde{\mathbf{W}}_0 = \mathbf{W}, \tag{17}$$

where $\widetilde{\mathbf{W}}$ denotes the trainable latent weights initialized from the FP weights $\mathbf{W}$, and $\mathbf{s} \in \mathbb{R}_{>0}^{c_{out} \times \frac{c_{in}}{g}}$ denotes the group-wise quantization scales. Unlike FlexRound and OmniQuant, which keep the FP weights fixed and optimize only auxiliary quantization variables, Block-AP jointly optimizes $\phi_{\text{Block-AP}} = \{\widetilde{\mathbf{W}}, \mathbf{s}\}$ during block-wise reconstruction.

[5] Following the integration setting of LFQ (Lee et al., 2026), we consider OmniQuant with LWC only.

B Predictive Sensitivity of Block-Output Perturbations

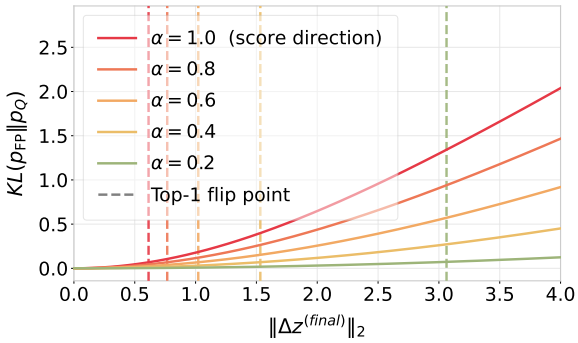


Figure 5: KL divergence between the FP predictive distribution and the distribution induced by final-block perturbations on Qwen2.5-0.5B. The perturbations vary in magnitude and direction, with α denoting the cosine similarity between the perturbation and score directions. Vertical dashed lines indicate the perturbation magnitudes at which the top-1 prediction changes.

To illustrate the directional sensitivity of the predictive distribution, we perturb the output of the final Transformer block along directions with different degrees of alignment to the score direction. Specifically, α denotes the cosine similarity between the perturbation and score directions, where $\alpha = 1$ indicates full alignment and $\alpha = 0$ indicates orthogonality.

As shown in Fig. 5, perturbations with stronger alignment to the score direction induce substantially larger KL divergence for the same perturbation magnitude. They also cause the top-1 prediction to change at smaller perturbation magnitudes, as indicated by the vertical dashed lines. These observations illustrate that equal-magnitude block-output errors can have markedly different effects on the predictive distribution depending on their directions. This directional disparity motivates us to assign greater reconstruction weight to perturbations aligned with predictively sensitive score directions.

C Geometric Interpretation of the Regularized Objective

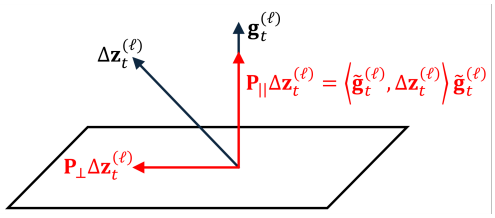


Figure 6: Geometric decomposition of the block-output perturbation $\Delta\mathbf{z}_t^{(\ell)}$ into components parallel and orthogonal to the FP-derived score direction $\mathbf{g}_t^{(\ell)}$. The score-guided term assigns additional directional weight only to the parallel component.

For a nonzero score vector $\mathbf{g}_t^{(\ell)}$, we decompose the block-output perturbation into components parallel and orthogonal to the score direction, as illustrated in Fig. 6:

$$\Delta\mathbf{z}_t^{(\ell)} = \mathbf{P}_{\|}\Delta\mathbf{z}_t^{(\ell)} + \mathbf{P}_{\perp}\Delta\mathbf{z}_t^{(\ell)}, \tag{18}$$

where

$$\mathbf{P}_{\|}\Delta\mathbf{z}_t^{(\ell)} = \left\langle \frac{\mathbf{g}_t^{(\ell)}}{\|\mathbf{g}_t^{(\ell)}\|_2}, \Delta\mathbf{z}_t^{(\ell)} \right\rangle \frac{\mathbf{g}_t^{(\ell)}}{\|\mathbf{g}_t^{(\ell)}\|_2}. \tag{19}$$

The regularized token-wise objective can then be written as

$$\begin{aligned}
&\left\|\Delta\mathbf{z}_t^{(\ell)}\right\|_2^2 + \lambda_t^{(\ell)} \langle \mathbf{g}_t^{(\ell)}, \Delta\mathbf{z}_t^{(\ell)} \rangle^2 \\
&= \left(\left\|\mathbf{P}_{\perp}\Delta\mathbf{z}_t^{(\ell)}\right\|_2^2 + \left\|\mathbf{P}_{\|}\Delta\mathbf{z}_t^{(\ell)}\right\|_2^2 \right) + \lambda_t^{(\ell)} \left\|\mathbf{g}_t^{(\ell)}\right\|_2^2 \left\langle \frac{\mathbf{g}_t^{(\ell)}}{\left\|\mathbf{g}_t^{(\ell)}\right\|_2}, \Delta\mathbf{z}_t^{(\ell)} \right\rangle^2 \\
&= \left\|\mathbf{P}_{\perp}\Delta\mathbf{z}_t^{(\ell)}\right\|_2^2 + \left(1 + \lambda_t^{(\ell)} \left\|\mathbf{g}_t^{(\ell)}\right\|_2^2\right) \left\|\mathbf{P}_{\|}\Delta\mathbf{z}_t^{(\ell)}\right\|_2^2.
\end{aligned} \tag{20}$$

Thus, standard reconstruction controls perturbations in all directions equally, while the score-guided term assigns additional weight to the component parallel to the predictively sensitive score direction.

D EXPERIMENTAL SETUP

Model. We use the following four open-source LLM checkpoints, all obtained from the Hugging Face Hub: `Qwen/{Qwen3-4B-Instruct, Qwen3-8B, Qwen2.5-14B-Instruct}` and `meta-llama/Meta-Llama-3.1-8B-Instruct`. For Qwen3 models, we disable thinking mode by setting `enable_thinking=False` and apply the standard chat template associated with each model.

Calibration. We construct a mixed dataset comprising 384 samples from Alpaca (Taori et al., 2023) and 128 samples from C4 (Raffel et al., 2020), corresponding to a $3:1$ mixing ratio with a fixed sequence length of 2048 tokens. Each sequence is tokenized using the respective model's tokenizer without additional padding or truncation.

Score-vector computation. For each calibration sequence, we independently sample $v_s \sim p_{\mathrm{FP},s}$ once per predictive position and compute $\mathbf{g}_t^{(\ell)} = \nabla_{\mathbf{z}_t^{(\ell)}} \sum_{s=t}^{T} \log p_{\mathrm{FP},s}(v_s)$. The score vectors for all blocks are obtained from the same FP forward and backward pass. We set $\tau^{(\ell)}$ to the median nonzero value of $\|\mathbf{g}_t^{(\ell)}\|_2^2$ over the 512×2048 calibration tokens. For Qwen3-4B-Instruct, $\tau^{(\ell)}$ ranges from 2.2×10^{-7} at ℓ=0 to 8.7×10^{-10} at ℓ=35. The normalized vectors $\mathbf{g}_t^{(\ell)} / \sqrt{\tau + \|\mathbf{g}_t^{(\ell)}\|_2^2}$ are computed once, cached in bfloat16 separately for each block, and kept fixed throughout block-wise reconstruction. For ablation variants requiring unnormalized score vectors, we cache the raw score vectors separately.

Optimization. We reconstruct the Transformer blocks sequentially, using the quantized output of the preceding block as input. Each block is optimized for 5000 iterations with a mini-batch size of 4. We follow the learning-rate and optimization settings of each base quantization method.

Hyperparameter selection. For each model, base method, and quantization setting, we select (λ, ρ) primarily based on WikiText2 PPL, using mean downstream accuracy to break near-ties. Selection uses the reported WikiText2 and downstream evaluation sets. One selected checkpoint is used for all metrics in each row, with (λ, ρ) listed in Table 4. Every (λ, ρ) pair plotted in Fig. 4(a) improves all reported metrics over MSE.

Table 4: **Selected final-block directional strength λ and depth-decay factor ρ.** Each entry reports λ/ρ for the corresponding base method, model, and quantization setting.

Backend	Bits	Llama-3.1-8B	Qwen3-8B	Qwen3-4B	Qwen2.5-14B
Block-AP	W2g128	$3{\cdot}10^3$ / 0.93	$3{\cdot}10^3$ / 0.97	10^3 / 0.94	$3{\cdot}10^3$ / 1.0
	W3	$3{\cdot}10^3$ / 0.965	10^3 / 0.94	10^3 / 0.97	10^3 / 0.975
	W3g128	10^3 / 0.93	300 / 0.97	10^3 / 0.94	300 / 0.975
OmniQuant	W3	$3{\cdot}10^3$ / 0.93	$3{\cdot}10^3$ / 0.97	$3{\cdot}10^3$ / 0.94	10^3 / 0.975
	W3g128	$3{\cdot}10^3$ / 0.93	$3{\cdot}10^3$ / 0.94	$3{\cdot}10^3$ / 0.94	$3{\cdot}10^3$ / 0.95
FlexRound	W2g128	$3{\cdot}10^3$ / 0.93	$2{\cdot}10^3$ / 1.0	200 / 0.94	10^4 / 1.0
	W3	$3{\cdot}10^3$ / 0.93	300 / 0.925	10^3 / 0.97	300 / 0.95
	W3g128	300 / 0.93	10^3 / 0.97	$3{\cdot}10^3$ / 0.94	300 / 0.95

Notes. The selected values span $\lambda \in [200, 10^4]$ and $\rho \in [0.925, 1]$. Llama-3.1-8B, Qwen3-4B, and Qwen2.5-14B denote their instruction-tuned variants. OmniQuant is not evaluated under W2g128.

E Component Ablation Protocol

Setup. We conduct the component ablations on Qwen3-4B-Instruct (L=36 blocks, d=2560) with Block-AP from EfficientQAT under symmetric W3g128 and W2g128 weight quantization. All rows use the same 512 calibration sequences of length 2048, sequence order, underlying score vectors, and optimization seed 42. The MSE baseline uses the standard Block-AP reconstruction objective $\sum_t \|\Delta \mathbf{z}_t^{(\ell)}\|_2^2$.

Objective. Except for the direction-only variant, all rows optimize the regularized objective in (7), averaged over the tokens in each mini-batch. We use the depth-dependent strength $\lambda^{(\ell)} = \lambda \rho^{L-1-\ell}$ with λ=10^3, ρ=0.94, yielding $\lambda^{(0)} = 114.7$, $\lambda^{(35)} = 10^3$. The full objective uses $\lambda_t = \lambda^{(\ell)}/(\tau + \|\mathbf{g}_t^{(\ell)}\|_2^2)$ as defined in (10). Each ablation modifies only the magnitude regularizer, the directional coefficient, or the score structure used in the directional term.

Variants.

- *Direction term only*: We remove the MSE term from (12), keeping its normalized directional penalty unchanged. Although the optimized directional loss decreases normally, the unoptimized MSE loss grows substantially under W3g128, reaching 1.3×10^4 at $\ell = 10$ and 1.2×10^5 at $\ell = 35$, compared with 66 and 2.5×10^4, respectively, under the full objective.
- *Unnormalized, original strength*: We set $\lambda_t^{(\ell)} = \lambda^{(\ell)}$ and use the raw score vector $\mathbf{g}_t^{(\ell)}$. The additional score-parallel weight is therefore $\lambda^{(\ell)}\|\mathbf{g}_t^{(\ell)}\|_2^2$. Because the squared gradient norms are typically on the order of 10^{-8} to 10^{-10}, this directional contribution is extremely small at λ=10^3. Consequently, this variant behaves similarly to standard MSE reconstruction.
- *Block-wise scaling only*: We set $\lambda_t^{(\ell)} = \lambda^{(\ell)}/\tau^{(\ell)}$ and use the raw score vectors. This configuration corrects the block-level gradient scale but omits the token-dependent term $\|\mathbf{g}_t^{(\ell)}\|_2^2$ from the denominator in (10). Its additional score-parallel weight becomes $\lambda^{(\ell)}\|\mathbf{g}_t^{(\ell)}\|_2^2/\tau^{(\ell)}$. This weight equals $\lambda^{(\ell)}$ at the median-gradient token but is not bounded independently of the token gradient norm. Consequently, upper-tail tokens in Fig. 2(b) can receive weights several orders of magnitude larger than the median.
- *Token-shared Fisher*: We use the calibration-token average of normalized score outer products, $\bar{\mathbf{A}}^{(\ell)} = \mathbb{E}_t[\mathbf{g}_t^{(\ell)}(\mathbf{g}_t^{(\ell)})^T/(\tau^{(\ell)} + \|\mathbf{g}_t^{(\ell)}\|_2^2)]$, in the directional penalty $\lambda^{(\ell)}(\Delta \mathbf{z}_t^{(\ell)})^T \bar{\mathbf{A}}^{(\ell)} \Delta \mathbf{z}_t^{(\ell)}$. The same matrix is shared across token positions.
- *Shared mean score vector*: We replace every token-specific score vector in block ℓ with a single vector $\bar{\mathbf{g}}^{(\ell)}$ averaged over the current mini-batch. The directional coefficient is then evaluated using $\bar{\mathbf{g}}^{(\ell)}$. Unlike the token-shared Fisher variant, this construction yields a single rank-one direction shared by all token positions in that mini-batch.
- *Token-shuffled score vectors*: Within each mini-batch, we randomly permute the nonzero-gradient token positions using a permutation π and replace $\mathbf{g}_t^{(\ell)}$ with $\|\mathbf{g}_t^{(\ell)}\|_2 \, \mathbf{g}_{\pi(t)}^{(\ell)}/\|\mathbf{g}_{\pi(t)}^{(\ell)}\|_2$. This operation assigns token t the score direction of another token while preserving its original gradient norm and directional coefficient. The permutation is redrawn at each optimization step using a block-specific generator with seed $10007 + \ell$. This control isolates token-score correspondence without altering the marginal distribution of gradient norms.
- *Coordinate-diagonal approximation*: We replace $\langle \mathbf{g}_t^{(\ell)}, \Delta \mathbf{z}_t^{(\ell)} \rangle^2$ with $\sum_k (g_{t,k}^{(\ell)})^2 (\Delta z_{t,k}^{(\ell)})^2$. This approximation retains the diagonal of $\mathbf{g}_t^{(\ell)}(\mathbf{g}_t^{(\ell)})^T$ while setting all cross-feature terms to zero. The token-dependent coefficient $\lambda_t^{(\ell)}$ remains unchanged.

F Effect of the Number of Monte Carlo Samples

In the main experiments, we used a single label sample per token to obtain a stochastic rank-one approximation of the position-wise Fisher matrix. To examine whether using additional samples improves quantization performance, we consider $M \in \{1, 2, 4\}$ labels independently sampled from the FP predictive distribution for each token. The resulting score outer products provide a rank-at-most-M Monte Carlo estimate of the position-wise Fisher matrix, and the corresponding directional loss terms are averaged during reconstruction.

As shown in Table 5, all evaluated values of M outperform standard MSE reconstruction in PPL and nearly all downstream evaluations. Although using multiple samples can provide a higher-rank Fisher approximation and further improve PPL, the downstream benefits are limited and configuration-dependent. These results suggest that a single sampled score vector already captures useful information about predictively sensitive directions. Moreover, increasing M requires computing, storing, and evaluating multiple score vectors for every token. We therefore used $M = 1$ throughout the main experiments for its favorable accuracy, memory, and runtime trade-off.

Table 5: **Effect of the number of Monte Carlo samples M.** Results on Qwen3-4B-Instruct with Block-AP under W3g128 and W2g128 quantization ($\lambda = 10^3, \rho = 0.94$). **Bold** indicates the best result among $M \in \{1, 2, 4\}$.

Loss	Bits	PPL↓	CSR6↑	MMLU↑	IFEval↑
MSE	W3g128	13.32	69.6	66.3	70.1
SABRE ($M = 1$)	W3g128	12.43	70.0	66.7	**73.9**
SABRE ($M = 2$)	W3g128	12.07	**70.1**	**67.3**	66.2
SABRE ($M = 4$)	W3g128	**12.05**	69.8	67.2	70.6
MSE	W2g128	40.82	57.4	48.9	40.3
SABRE ($M = 1$)	W2g128	29.30	59.6	50.9	**43.3**
SABRE ($M = 2$)	W2g128	29.33	**60.2**	**51.8**	41.8
SABRE ($M = 4$)	W2g128	**27.50**	60.0	50.7	42.5

G Comparison against Training-Free Second-Order PTQ Methods

Our main experiments evaluate SABRE as an objective-level extension to existing block-wise quantization methods. For broader comparison, we evaluate the resulting Block-AP and FlexRound integrations against GPTQ and GuidedQuant under a shared symmetric-uniform quantization protocol. We use the official implementations of both baselines with GuidedQuant incorporating end-loss saliency into GPTQ. Our comparison places all methods under the same low-bit quantization and calibration settings rather than reproducing their individually optimized configurations.

As shown in Table 6, the block-wise integrations of SABRE generally achieve better performance than GPTQ and GuidedQuant across W2g128, W3, and W3g128 quantization. The gains are particularly pronounced under W2g128, where FlexRound+SABRE reduces PPL from 65.78 with GuidedQuant to 23.44 and improves IFEval from 25.9% to 45.5%. Under W3 and W3g128, FlexRound+SABRE achieves the best performance across all reported metrics. These results show that the additional optimization cost of score-guided block-wise reconstruction can yield substantial performance gains over computationally efficient, training-free second-order methods, particularly under aggressive low-bit quantization.

Table 6: **Comparison against training-free second-order PTQ baselines on Qwen3-8B.** Within each quantization setting, the best result is shown in **bold**. For W2g128, GSM8K and MATH500 are omitted because all methods achieve near-zero performance.

(a) W2g128

Method	PPL↓	CSR6↑	MMLU↑	IFEval↑
GPTQ	64.83	47.2	42.2	8.7
GPTQ + GuidedQuant	65.78	58.5	42.0	25.9
Block-AP + SABRE (ours)	26.05	64.1	50.3	28.5
FlexRound + SABRE (ours)	**23.44**	**65.8**	**56.6**	**45.5**

(b) W3 and W3g128

		Modeling / understanding			Text generation		
Method	Bits	PPL↓	CSR6↑	MMLU↑	IFEval↑	GSM8K↑	MATH500↑
GPTQ	W3	35.70	55.7	56.0	24.8	36.6	6.2
GPTQ + GuidedQuant	W3	17.22	65.0	58.2	51.8	58.5	25.4
Block-AP + SABRE (ours)	W3	12.43	73.2	69.7	68.2	74.7	56.6
FlexRound + SABRE (ours)	W3	**11.76**	**73.9**	**71.2**	**76.5**	**81.7**	**63.8**
GPTQ	W3g128	14.35	70.3	68.5	73.2	81.2	59.2
GPTQ + GuidedQuant	W3g128	11.82	70.9	69.4	75.8	81.1	62.4
Block-AP + SABRE (ours)	W3g128	11.11	72.6	70.6	75.8	77.5	61.2
FlexRound + SABRE (ours)	W3g128	**10.24**	**73.6**	**72.2**	**78.6**	**84.8**	**68.2**

H ADDITIONAL EXPERIMENTS

H.1 ADDITIONAL 3-BIT INTEGRATION RESULTS

We further evaluate the integration of SABRE with Block-AP, OmniQuant, and FlexRound on Llama-3.1-8B-Instruct under per-channel W3 and group-wise W3g128 quantization. As shown in Table 7, SABRE reduces PPL across all evaluated base methods and quantization granularities. The downstream results are broadly favorable, with SABRE maintaining or improving performance across most evaluated configurations. These results provide additional evidence that the benefits of SABRE extend beyond the Qwen model family evaluated in the main 3-bit experiments.

Table 7: **Additional 3-bit integration results on Llama-3.1-8B-Instruct.** Within each pair, the better result is shown in **bold**. SABRE reduces PPL across all evaluated base methods and quantization granularities while maintaining or improving most downstream results.

		Modeling / understanding			Text generation		
Method	Bits	PPL↓	CSR6↑	MMLU↑	IFEval↑	GSM8K↑	MATH500↑
Full precision	FP16	6.75	75.9	68.2	73.8	84.8	48.0
Block-AP	W3	9.54	73.2	61.2	62.8	**68.6**	**28.4**
+ SABRE (ours)	W3	**9.20**	**73.6**	**62.0**	62.8	65.4	26.4
Block-AP	W3g128	8.40	73.5	63.0	61.9	**71.9**	28.0
+ SABRE (ours)	W3g128	**8.35**	**73.9**	**63.7**	**62.5**	69.3	**30.2**
OmniQuant	W3	32.83	69.1	53.7	47.3	30.9	9.6
+ SABRE (ours)	W3	**24.95**	**70.2**	**54.0**	**52.3**	**35.2**	**11.8**
OmniQuant	W3g128	10.53	70.9	60.6	**64.0**	58.3	22.8
+ SABRE (ours)	W3g128	**9.59**	**72.0**	**61.2**	62.3	**64.1**	**23.4**
FlexRound	W3	9.47	**74.3**	63.9	66.9	71.3	30.2
+ SABRE (ours)	W3	**9.18**	74.1	**64.0**	66.9	**72.1**	**31.6**
FlexRound	W3g128	7.91	74.0	**64.5**	69.7	76.1	33.8
+ SABRE (ours)	W3g128	**7.87**	**74.7**	64.3	**70.2**	**78.5**	**36.6**

H.2 AGREEMENT WITH THE FULL-PRECISION MODEL

We further evaluate whether SABRE better preserves the predictive behavior of the FP model on held-out data. We use Qwen3-4B-Instruct with Block-AP under W2g128 quantization and evaluate on a held-out C4 validation split that is disjoint from both the calibration data and all evaluation benchmarks.

As shown in Table 8, SABRE reduces the KL divergence from the FP predictive distribution from 0.946 to 0.837 and lowers the token-level cross-entropy from 3.642 to 3.459. It also improves top-1 agreement with the FP model from 65.6% to 66.9%. These consistent improvements indicate that the score-guided objective better preserves the FP model's predictive behavior on unseen text, providing direct empirical support for the predictive-sensitivity motivation of SABRE.

Table 8: **Agreement with the FP model on a held-out C4 validation split.** Results on Qwen3-4B-Instruct with Block-AP under W2g128 quantization ($\lambda = 10^3$, $\rho = 0.94$). The validation split is disjoint from the calibration data and evaluation benchmarks. KL denotes the KL divergence from the FP predictive distribution, CE denotes token-level cross-entropy, and Agree denotes top-1 prediction agreement with the FP model.

Method	KL↓	CE↓	Agree↑
Full precision (FP16)	0.000	2.706	100.0
Block-AP (MSE)	0.946	3.642	65.6
+ SABRE (λ=10^3, ρ=0.94)	**0.837**	**3.459**	**66.9**

H.3 Effectiveness of Gradient-Norm Normalization

As described in (9), without the token-dependent normalization term in (10), the additional score-parallel weight remains proportional to $\|\mathbf{g}_t^{(\ell)}\|_2^2$ and is therefore not bounded independently of the gradient norm. We examine the effect of rescaling unnormalized directional weighting. Specifically, we rescale the unnormalized term so that its block-wise mean token weight matches that of SABRE, and additionally evaluate $0.3\times$ and $3\times$ this matched strength.

As shown in Table 9, SABRE achieves the best downstream scores at both bit-widths and the lowest W2g128 PPL. The $0.3\times$ and $1\times$ variants yield slightly lower W3g128 PPL, while all three unnormalized variants yield lower W2g128 KL. The proposed token-dependent normalization bounds the directional weight and limits the influence of large-gradient tokens.

Table 9: **Effectiveness of gradient-norm normalization.** Results on Qwen3-4B-Instruct with Block-AP under W3g128 and W2g128 quantization (λ=10^3, ρ=0.94). The unnormalized directional term is rescaled to $0.3\times$, $1\times$, and $3\times$ the block-wise mean token weight of SABRE, where $1\times$ denotes the mean-matched setting. KL is evaluated on 128 held-out C4 validation sequences; bold marks the best value in each column.

	W3g128					W2 g128				
Variant	PPL↓	KL↓	CSR6↑	MMLU↑	IFEval↑	PPL↓	KL↓	CSR6↑	MMLU↑	IFEval↑
Block-AP (MSE)	13.32	0.127	69.6	66.3	70.1	40.82	0.946	57.4	48.9	40.3
Unnormalized ($\times$0.3)	**12.34**	0.136	68.8	66.2	73.8	34.04	**0.587**	57.8	47.6	40.9
Unnormalized ($\times$1)	12.37	0.155	68.1	65.1	71.3	36.98	0.671	56.9	42.6	34.6
Unnormalized ($\times$3)	13.02	0.195	68.3	63.0	68.8	34.87	0.793	54.7	36.3	26.1
SABRE (ours)	12.43	**0.121**	**70.0**	**66.7**	**73.9**	**29.30**	0.837	**59.6**	**50.9**	**43.3**

H.4 Robustness to Optimization Seeds

We evaluate the robustness of SABRE across five optimization seeds while keeping the calibration set fixed. As shown in Table 10, SABRE improves PPL, MMLU, and IFEval across all five seeds and CSR6 in four of the five seeds. On average, SABRE improves CSR6, MMLU, and IFEval by 1.6, 1.5, and 2.9 percentage points, respectively. These results demonstrate that the improvements of SABRE are robust to optimization-seed variation rather than being driven by a favorable individual run.

Table 10: **Robustness to optimization seeds.** Results on Qwen3-4B-Instruct with Block-AP under W2g128 quantization (λ=10^3, ρ=0.94). The calibration set is fixed across runs. The upper rows report means and sample standard deviations over seeds 42–46, and the lower rows report the per-seed difference between SABRE and MSE.

Method	PPL↓	CSR6↑	MMLU↑	IFEval↑
Block-AP	50.64±7.71	58.1±0.9	49.1±0.7	41.3±1.3
+ SABRE (ours)	**32.98**±**4.33**	**59.7**±**0.5**	**50.6**±**0.3**	**44.2**±**1.2**
Per-seed difference, SABRE − MSE				
seed 42	-11.52	+2.2	+2.0	+3.0
seed 43	-26.93	-0.7	+0.4	+5.4
seed 44	-13.95	+1.6	+1.5	+2.4
seed 45	-28.53	+2.8	+1.5	+1.3
seed 46	-7.33	+2.1	+2.0	+2.6

H.5 Robustness to Calibration Data

Our main experiments use a calibration mixture consisting of Alpaca and C4. To examine whether the effectiveness of SABRE depends on this mixture, we additionally conduct calibration using only C4 sequences. We use Block-AP with 512 calibration sequences and otherwise retain the main experimental settings, except where noted in the table caption.

As shown in Table 11, SABRE improves PPL, CSR6, MMLU, and IFEval across all evaluated model and quantization settings. The improvements are particularly pronounced under W2g128, where SABRE reduces PPL from 30.34 to 23.43 on Qwen3-4B-Instruct and from 97.85 to 20.45 on Qwen3-8B. Generation performance also generally improves, with only minor decreases on Qwen3-8B under W3g128. These results indicate that the benefits of score-guided reconstruction are not specific to the mixed calibration dataset used in the main experiments and remain effective with C4-only calibration.

Table 11: **Results with C4-only calibration using Block-AP.** All other settings follow the main experiments. Within each pair, the better result is shown in **bold**. SABRE uses $\lambda=10^3$ and $\rho=0.94$ in every cell except Qwen3-4B-Instruct W3 ($\rho=0.97$) and Qwen3-8B W2g128 ($\lambda=3\cdot10^3$, $\rho=0.97$). For W2g128, GSM8K and MATH500 are omitted because performance remains near zero.

(a) W2g128

Method	PPL↓	CSR6↑	MMLU↑	IFEval↑
Qwen3-4B-Instruct				
Block-AP	30.34	57.2	40.5	22.4
+ SABRE (ours)	**23.43**	**58.7**	**45.1**	**29.0**
Qwen3-8B				
Block-AP	97.85	60.4	36.7	10.4
+ SABRE (ours)	**20.45**	**61.4**	**43.6**	**10.9**

(b) W3 and W3g128

		Modeling / understanding			Text generation		
Method	Bits	PPL↓	CSR6↑	MMLU↑	IFEval↑	GSM8K↑	MATH500↑
Qwen3-4B-Instruct							
Block-AP	W3	13.98	68.5	64.1	69.7	79.2	43.4
+ SABRE (ours)	W3	**13.27**	**69.6**	**66.0**	**70.2**	**80.5**	**53.0**
Block-AP	W3g128	12.09	69.7	65.6	68.6	81.9	58.8
+ SABRE (ours)	W3g128	**11.70**	**70.2**	**66.3**	**72.6**	**83.0**	**59.4**
Qwen3-8B							
Block-AP	W3	11.96	72.4	68.8	48.6	68.3	39.6
+ SABRE (ours)	W3	**11.61**	**72.9**	**69.5**	**66.0**	**72.6**	**50.4**
Block-AP	W3g128	10.76	72.8	69.8	70.6	**80.0**	**58.8**
+ SABRE (ours)	W3g128	**10.48**	**72.9**	**70.2**	**75.2**	78.9	58.4

H.6 Results on Reasoning-Oriented Models

We further evaluate SABRE on two reasoning-oriented models, L1-Qwen-7B-Max (Aggarwal & Welleck, 2025) and DeepSeek-R1-Distill-Llama-8B (Guo et al., 2025), using Block-AP under W3g128 and W4g128 quantization. For each model, we evaluate all quantized variants using identical prompts, maximum generation lengths, and decoding configurations.

As shown in Table 12, SABRE improves MATH500 and AIME'24 across all evaluated model-bit-width pairs. The gains are particularly pronounced on AIME'24: under W3g128, pass@1 increases from 26.88% to 33.33% for L1-Qwen-7B-Max and from 23.54% to 32.50% for DeepSeek-R1-Distill-Llama-8B. Under W4g128, SABRE further improves AIME'24 pass@1 from 41.88% to 53.96% and MATH500 from 68.38% to 71.03% on DeepSeek-R1-Distill-Llama-8B. These results provide additional evidence that score-guided reconstruction can benefit reasoning-oriented models and mathematical reasoning tasks.

Table 12: **Quantization results on reasoning-oriented models.** For sampled evaluation, we use a temperature of 0.6 and top-p of 0.95, with 16 responses per question for AIME'24 pass@1. The maximum generation length is 4096 tokens for MATH500. For AIME'24, we use 8192 tokens for greedy decoding and L1-Qwen-7B-Max pass@1, and 16384 tokens for DeepSeek-R1-Distill-Llama-8B pass@1. Within each model and quantization setting, the better result is shown in **bold**.

		L1-Qwen-7B-Max				DeepSeek-R1-Distill-Llama-8B			
Method	Bits	PPL↓	MATH500 (avg@8)	AIME'24 (greedy)	AIME'24 (pass@1)	PPL↓	MATH500 (avg@8)	AIME'24 (greedy)	AIME'24 (pass@1)
Block-AP	W3g128	33.12	83.40	23.33	26.88	14.26	59.38	13.33	23.54
+ SABRE	W3g128	**30.14**	**84.60**	**30.00**	**33.33**	**13.53**	**61.48**	**16.67**	**32.50**
Block-AP	W4g128	31.02	87.15	43.33	49.17	**12.41**	68.38	26.67	41.88
+ SABRE	W4g128	**29.92**	**88.13**	**46.67**	**50.00**	12.45	**71.03**	**33.33**	**53.96**