CET: Spectral Geometry for Mixed-Precision Quantization

Anonymous Author(s)

Abstract

As deep models continue to scale, post-training quantization has become an important tool for efficient deployment. Compared with uniform quantization, mixed-precision quantization offers greater compression potential by assigning different bit-widths according to the quantization tolerance of different layers. However, existing methods mainly rely on layer-wise sensitivity ranking or discrete search for bit allocation, while making limited use of the directional differences of quantization perturbations in the local loss landscape. To address this limitation, we propose Compression Error Topography (CET), which models mixed-precision quantization as structured perturbations in parameter space and relates quantization perturbations to loss variation through a local second-order approximation. CET further leverages the spectral structure of the Hessian to characterize curvature variations across perturbation directions, identify low-curvature geometric subspaces that are more compression-friendly, and use them to guide layer-wise mixed-precision allocation. Extensive experiments across vision, language, and generative models demonstrate that CET achieves competitive compression–accuracy trade-offs, with particularly strong performance in mixed-precision and ultra-low-bit regimes.

CCS Concepts

• **Computing methodologies** → **Artificial intelligence**.

Keywords

Compression, Mixed-Precision Quantization, Algebraic Geometry

ACM Reference Format:

. 2026. CET: Spectral Geometry for Mixed-Precision Quantization. In *Proceedings of (Conference acronym 'XX)*. ACM, New York, NY, USA, 9 pages.

1 Introduction

The continued scaling of dense neural networks—including convolutional backbones for visual recognition, encoder transformers for language understanding, autoregressive language models, and diffusion backbones for generative modeling—has substantially improved model quality, while making deployment increasingly constrained by model footprint, memory bandwidth, and latency. These pressures are particularly acute in small-batch interactive serving [16], privacy-sensitive [8] or on-device inference, and heterogeneous production environments [17, 25], where predictable computation and mature low-bit kernel support remain critical. Consequently, much recent work has focused on compressing pretrained dense checkpoints into deployable low-bit models. In this setting, the key challenge is not uniform precision reduction, but precision allocation, since layers within the same model can differ sharply in their tolerance to compression. This makes mixed-precision quantization especially appealing for deployment-oriented compression.

Despite this promise, the central problem in mixed-precision quantization remains how to allocate layer-wise bit-widths in a principled way. Existing methods address this problem through layer-wise sensitivity ranking, proxy criteria, or discrete search. Early Hessian-based approaches such as HAWQ and HAWQ-V2 [6, 7] showed that second-order information is useful for layer-wise precision assignment, while more recent methods extended mixed-precision quantization to LLMs, vision transformers, and hybrid compression settings [2, 4, 18, 23]. However, most approaches ultimately reduce the allocation problem to scalar layer scores, greedy selection, or reconstruction-based criteria. As a result, they answer which layers are more sensitive, but make limited use of a complementary question that is equally important for quantization: which perturbation directions are better tolerated by the loss landscape.

This distinction becomes especially important in aggressive mixed-precision and ultra-low-bit settings. In these regimes, the main difficulty is not simply precision assignment, but sensitivity mischaracterization. Existing methods [3, 4, 6, 13] typically compress second-order information into layer-wise scalar sensitivity estimates and then rank or search accordingly. Yet this scalar view is often insufficient: our empirical diagnostics show that different sensitivity-based methods can produce conflicting judgments about layer fragility under the same compression regime, while even identical nominal precision settings can lead to different loss increases once the realized quantization errors differ. These observations suggest that the bottleneck lies in the sensitivity model itself. Quantization error is not harmful only because of how much of it is introduced, but because of how it is oriented relative to the local loss landscape. As a result, sensitivity estimation that ignores perturbation direction can still push the model toward steeper local geometry and yield avoidable degradation.

Motivated by this observation, we propose Compression Error Topography (CET), a geometry-driven framework for mixed-precision quantization. CET models quantization as structured perturbations in parameter space and relates quantization-induced loss variation to a local second-order approximation around the pretrained model. This formulation yields a geometric interpretation through the Hessian spectrum: different perturbation directions correspond to different local curvatures, and perturbations aligned with flatter local geometry induce smaller loss increases than those aligned with steeper directions. CET leverages this directional anisotropy to identify low-curvature geometric subspaces that are more compression-friendly, and uses them to guide layer-wise mixed-precision allocation. In this sense, CET does not treat Hessian information merely as a sensitivity proxy, but uses spectral geometry to explain how perturbations behave locally and how this behavior should shape precision assignment.

Extensive experiments show that this geometric perspective is effective across heterogeneous model families. In the current draft, CET achieves strong compression–accuracy trade-offs on CNNs, BERT-style transformers, LLM benchmarks, and diffusion

Conference acronym 'XX,

or text-to-image generation tasks, with especially clear advantages in mixed-precision and ultra-low-bit regimes.

Our contributions are summarized as follows:

(1) We propose Compression Error Topography (CET), a geometry-driven framework for mixed-precision quantization. CET models quantization as structured perturbations in parameter space and relates loss variation to local second-order geometry.

(2) We introduce a spectral-geometric perspective for layer-wise bit allocation. By leveraging the Hessian spectrum, CET characterizes curvature differences across perturbation directions, identifies compression-friendly low-curvature subspaces, and uses them to guide mixed-precision assignment.

(3) Experiments on vision, language, and generative models show that CET achieves competitive compression–accuracy trade-offs, especially in mixed-precision and ultra-low-bit regimes.

2 Related Work

2.1 Post-Training Quantization

Post-training quantization (PTQ) is a widely used approach for reducing model size and inference cost without full retraining, and has been extended from conventional CNNs to large language models and vision transformers. As quantization moves into the low-bit regime, the main challenge is no longer only how to discretize weights and activations, but how to control the resulting quantization error so that model behavior remains stable after compression.

Recent PTQ methods mainly improve low-bit robustness through better local error control. QuIP [2] introduces incoherence processing for ultra-low-bit LLM quantization by combining adaptive rounding with random orthogonal transformations, while QuIP# [23] further improves this line with randomized Hadamard transforms for more accurate extreme compression. QLoRA [5] instead focuses on memory-efficient 4-bit quantization for finetuning. Beyond language models, PTQ4DM [21] extends post-training quantization to diffusion models for the multi-timestep denoising process. Taken together, these approaches substantially improve low-bit PTQ by reducing approximation error at the layer or block level.

However, once different layers are allowed to use different precisions, the problem shifts from local quantizer design to global precision allocation. In this setting, the central question is not only how to quantize an individual layer accurately, but how to assign different bit-widths across the network in a principled way. This transition naturally leads to mixed-precision quantization.

2.2 Mixed-Precision Quantization

Mixed-precision quantization is motivated by the observation that different layers respond differently to quantization, making uniform bit-width assignment often suboptimal. Early Hessian-based methods such as HAWQ and HAWQ-V2 [6, 7] introduced second-order information for layer-wise precision selection: HAWQ uses the top Hessian eigenvalue to measure layer sensitivity, while HAWQ-V2 argues that the average Hessian trace is a better sensitivity metric and combines it with Pareto-frontier search for automatic bit allocation. Subsequent methods [8, 22, 28, 32] extend this sensitivity-based paradigm to broader settings. GPTQ [9] uses approximate second-order information to perform one-shot low-bit quantization, compensating quantization error through a Hessian-informed update procedure. VPTQ [18] formulates extreme low-bit vector quantization with explicit second-order optimization, and further refines weights through channel-independent second-order updates. RSAVQ [27] uses Fisher-information-induced Riemannian to guide both error projection and sensitivity-aware bit allocation.

These methods demonstrate that non-uniform precision allocation is effective across modern architectures. However, most of them still use second-order information mainly as layer-wise scalar sensitivity proxies for ranking, search, or reconstruction. In contrast, CET uses the Hessian spectrum to model the directional geometry of quantization perturbations and derives mixed-precision allocation from geometry-favorable perturbation subspaces rather than scalar sensitivity scores.

3 Error Theoretical Analysis Framework

3.1 Perturbation and Loss

We begin by viewing mixed-precision quantization as introducing structured perturbations into the parameters of a pretrained model. Let w denote the full-precision parameters, and let $Q_{\mathbf{b}}(w)$ denote the quantized parameters under a layer-wise bit-width configuration $\mathbf{b} = \{b_1, \ldots, b_L\}$. The resulting quantization perturbation is $\delta^{(\mathbf{b})} = Q_{\mathbf{b}}(w) - w$. Under this notation, mixed-precision quantization can be analyzed through how the perturbation induced by $\mathbf{b}$ changes the model loss. Let $f(w)$ denote the expected loss of the pretrained model on dataset $\mathbb{D}$. For a quantized model, the loss change induced by $\mathbf{b}$ is $\Delta L(\mathbf{b}) = f\left(w + \delta^{(\mathbf{b})}\right) - f(w)$.

Directly optimizing this Equation over all discrete bit-width configurations is difficult. CET therefore introduces a local second-order surrogate around the pretrained solution to characterize how quantization perturbations affect the loss:

$$f(w+\delta) - f(w) \approx g^\top \delta + \frac{1}{2}\delta^\top \mathbb{H}\delta, \tag{1}$$

where $g = \nabla f(w)$ and $\mathbb{H} = \nabla^2 f(w)$ are the gradient and Hessian evaluated at the pretrained model.

For a well-trained model, the gradient term is typically small within a local neighborhood of the solution. In this regime, the loss variation is dominated by the quadratic term:

$$\Delta L(\delta) \approx \frac{1}{2}\delta^\top \mathbb{H}\delta. \tag{2}$$

Eq. 2 serves as the basic surrogate of CET. It shows that the impact of quantization is determined not only by the magnitude of the perturbation, but also by how the perturbation is aligned with the local loss geometry. In particular, perturbations of similar norm can induce very different loss changes when they enter the loss landscape along different directions. This directional dependence is the starting point of our geometric analysis.

3.2 Quadratic Geometry

Eq. 2 admits a direct geometric interpretation. Consider the set of perturbations that induce the same second-order loss increase:

$$\delta^\top \mathbb{H}\delta = c, \tag{3}$$

where c is a constant. This level set defines an equal-loss surface in the local neighborhood of the model. Under CET, mixed-precision quantization is therefore analyzed not only through perturbation magnitude, but through the geometry of these equal-loss surfaces.

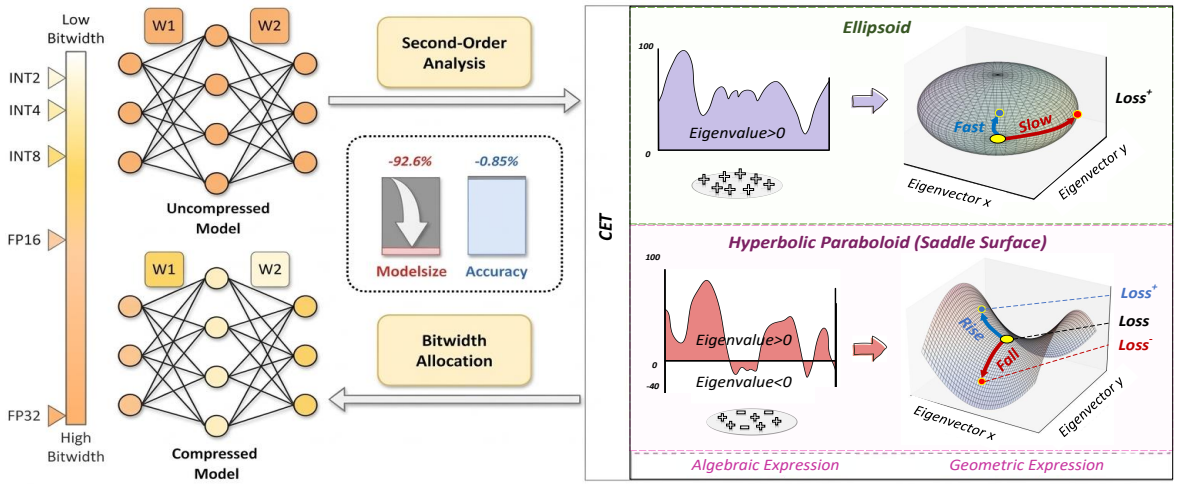

Figure 1: Overview of CET. Quantization is viewed as a parameter-space perturbation, and its loss effect is analyzed through local Hessian geometry. Flatter perturbation directions incur smaller loss increases and therefore guide layer-wise mix allocation.

Let the eigendecomposition of the Hessian be

$$\mathbb{H} = P\Lambda P^\top, \tag{4}$$

where $P = [p_1, \ldots, p_n]$ contains the eigenvectors of $\mathbb{H}$ and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ contains the corresponding eigenvalues. Writing the perturbation in the Hessian eigenbasis as $\delta = Py$, Eq. 2 becomes

$$\Delta L(\delta) \approx \frac{1}{2} y^\top \Lambda y = \frac{1}{2} \sum_{i=1}^{n} \lambda_i y_i^2. \tag{5}$$

Eq. 5 makes the directional nature of quantization-induced loss variation explicit. The eigenvectors define the principal perturbation directions, while the eigenvalues measure the local curvature along these directions. As a result, even perturbations with similar projected magnitude can incur very different loss increases when aligned with different eigendirections: directions associated with smaller eigenvalues are locally flatter and therefore less costly than directions associated with larger eigenvalues. This directional anisotropy is the geometric basis of CET.

This interpretation is illustrated in Fig. 1. After diagonalization, the equal-loss surface in Eq. 3 reduces to a canonical quadratic surface determined by the signs and magnitudes of λ_i. Two typical cases are particularly relevant.

(1) Ellipsoidal case. When the Hessian is positive definite, all eigenvalues are positive, and the level set forms an ellipsoidal surface whose principal axes are aligned with the Hessian eigenvectors. In this case, perturbations along different directions all increase loss, but at different rates: directions associated with larger eigenvalues lead to faster loss rise, while directions associated with smaller eigenvalues lead to slower loss rise. Equivalently, small eigenvalues correspond to longer axes and flatter local geometry, whereas large eigenvalues correspond to shorter axes and steeper geometry.

(2) Hyperbolic / saddle-like case. When the Hessian is indefinite, positive and negative eigenvalues coexist, so the level set is no longer closed and becomes hyperbolic in form. In low-dimensional intuition, this corresponds to a saddle-like local geometry. In this case, the effect of perturbations is again direction-dependent: along directions with positive curvature, the loss rises, whereas along directions with negative curvature, the loss falls. Therefore, even under the same perturbation magnitude, different directions can induce qualitatively different loss variations.

In CET, negative eigenvalues do not necessarily indicate harmful perturbations; under the local quadratic model, they may correspond to directions with smaller, or even negative, loss variation. For well-trained pretrained models, however, strongly negative directions are typically not dominant in the local spectrum.

3.3 Geometric Subspace

The directional analysis in Section 3.2 suggests a natural strategy for mixed-precision quantization: CET should favor perturbations aligned with geometry that induces smaller local quadratic cost, and suppress perturbations aligned with directions that cause large loss increase. In the ellipsoidal case, these are the long-axis directions. In the indefinite case, the geometry-favorable directions consist of all non-positive-curvature directions together with the positive-curvature directions of relatively small magnitude. CET therefore constructs a geometry-favorable subspace from these directions.

Because the Hessian eigenvectors form an orthogonal basis, the perturbation space can be decomposed as

$$\mathbb{R}^n = V_{\text{long}} \oplus V_{\text{short}}. \tag{6}$$

To define these subspaces precisely, let $\mathcal{I}_- = \{i : \lambda_i \le 0\}, \mathcal{I}_+ = \{i : \lambda_i > 0\}$. Partition the positive-curvature index set as $\mathcal{I}_+ = \mathcal{I}^+_{\text{long}} \cup \mathcal{I}_{\text{short}}$, such that $\lambda_i \le \lambda_j, \forall\, i \in \mathcal{I}^+_{\text{long}},\ j \in \mathcal{I}_{\text{short}}$. That is, $\mathcal{I}^+_{\text{long}}$ contains the smaller positive eigenvalues, while $\mathcal{I}_{\text{short}}$ contains the larger positive eigenvalues. CET then defines $V_{\text{long}} = \mathrm{span}\big(\{p_i : i \in \mathcal{I}_- \cup \mathcal{I}^+_{\text{long}}\}\big), V_{\text{short}} = \mathrm{span}\big(\{p_i : i \in \mathcal{I}_{\text{short}}\}\big)$. Hence, V_{long} contains all directions that do not induce large positive quadratic cost, while V_{short} contains the steeper positive-curvature directions that are most harmful under the local quadratic surrogate. In the theoretical analysis above, V_{long} and V_{short} are defined from the full eigendecomposition of the Hessian. In practice, CET approximates V_{short} using a truncated set of leading positive-curvature eigenvectors and treats the remaining directions as an approximation of V_{long}.

The following proposition formalizes this intuition.

Proposition 1. *Let $\mathbb{H} = P\Lambda P^\top$ be the eigendecomposition of the Hessian, and let V_{long} and V_{short} be defined as above. For any perturbations $u \in V_{\text{long}}$ and $v \in V_{\text{short}}$ with $\|u\|_2 = \|v\|_2$, the loss-increasing part of the quadratic variation satisfies*

$$\sum_{i \in \mathcal{I}_+} \lambda_i (P^\top u)_i^2 \;\le\; \sum_{i \in \mathcal{I}_+} \lambda_i (P^\top v)_i^2. \tag{7}$$

Moreover, if u has any component in $\mathcal{I}_-$, then the total quadratic variation $\frac{1}{2} u^\top \mathbb{H} u$ can be even smaller.

The result follows directly from Eq. 5. In the Hessian eigenbasis, the quadratic variation is a weighted sum of squared coordinates. Among positive-curvature directions, smaller eigenvalues induce no larger loss increase than larger eigenvalues under the same perturbation norm. Accordingly, in the positive-definite case, V_{long} reduces to the usual long-axis subspace, while in the indefinite case it additionally includes the non-positive-curvature directions that do not increase the loss-increasing part of the quadratic variation.

Proposition 1 provides the geometric foundation of CET. Under the same perturbation budget, perturbations aligned with V_{long} incur no larger local quadratic cost than perturbations aligned with V_{short}. CET therefore constructs a geometry-guided perturbation template by suppressing the perturbation energy on V_{short}, and then induces the mixed-precision assignment from this optimized geometric template.

Let $y = P^\top \delta$ denote the perturbation represented in the Hessian eigenbasis. Since $\mathcal{I}_{\text{short}} \subseteq \mathcal{I}_+$ contains only the larger positive-curvature directions, an ideal geometry-favorable perturbation should place little energy on this subspace:

$$\lambda_i y_i^2(\delta) \approx 0, \qquad i \in \mathcal{I}_{\text{short}}. \tag{8}$$

Eq. 8 is an ideal geometric condition rather than the final optimization problem. It expresses the principle that perturbation components aligned with the steeper positive-curvature directions should be suppressed as much as possible.

To obtain a non-trivial perturbation template while staying within the locality where the quadratic surrogate remains reliable, CET solves

$$\delta^\star = \arg\min_{\delta} \sum_{i \in \mathcal{I}_{\text{short}}} \lambda_i y_i^2(\delta) \quad \text{s.t. } \|\delta\|_2^2 = \epsilon. \tag{9}$$

The equality constraint fixes the perturbation budget, excludes the trivial zero solution, and matches Proposition 1, which compares perturbations under the same norm. In the idealized exact-subspace case, Eq. 9 characterizes perturbations that suppress the V_{short} component under a fixed norm budget. In practice, since V_{short} is approximated using a truncated set of leading positive-curvature eigenvectors, CET uses projected gradient descent with normalization as a practical numerical solver to construct a geometry-guided perturbation template. Therefore, PGD is used here as an implementation procedure. The resulting template fixes the perturbation magnitude and determines only its relative directional distribution under the local spectral geometry, while the final compression level is determined later by the calibration term C in Eq. 10.

3.4 Practical Bit Allocation with CET

CET instantiates the above geometric analysis into a practical mixed-precision bit-allocation procedure. Given a pretrained model and a calibration set, CET first uses a Lanczos-based spectral routine to estimate a truncated set of leading positive-curvature Hessian eigenpairs. These eigenvectors provide a practical approximation of the steep subspace V_{short}, while the remaining directions are treated as an approximation of the geometry-favorable subspace V_{long}. CET then initializes a perturbation $\delta^{(0)}$ under the prescribed norm budget and uses projected gradient descent with normalization as a practical numerical procedure to construct a geometry-guided perturbation template $\delta^\star$. This step suppresses perturbation components aligned with V_{short} under the truncated spectral approximation and yields a continuous geometry-guided template.

Algorithm 1 CET Algorithm for Mixed-Precision Quantization

Input: Pretrained model parameters w, calibration set $\mathbb{D}$, target compression budget B, bit range $[b_{\min}, b_{\max}]$
Output: Layer-wise bit-width assignment $\{b_1, \ldots, b_k\}$

1: **Estimate steep Hessian directions:** Use a Lanczos-based spectral routine on $\mathbb{D}$ to estimate the leading positive-curvature Hessian eigenpairs of $\mathbb{H}$.
2: **Construct geometric subspaces:** Use the retained leading positive-curvature eigenvectors to define V_{short}, and treat the remaining directions as the geometry-favorable subspace V_{long}.
3: **Initialize perturbation:** Randomly initialize $\delta^{(0)}$ in parameter space such that $\|\delta^{(0)}\|_2^2 = \epsilon$.
4: **Construct geometry-guided perturbation:** Solve Eq. 9 by projected gradient descent to obtain $\delta^\star$.
5: **Aggregate layer-wise geometric allowance:** For each layer i, aggregate the squared perturbation magnitude of $\delta^\star$ over the parameters of that layer to obtain δ_i.
6: **Generate candidate bit-widths:** Choose the calibration term C to satisfy the target budget B, and compute candidate precisions using Eq. 10.
7: **Candidate refinement:** For each layer, evaluate the rounded geometry-guided bit-width and one adjacent feasible discrete bit-width on $\mathbb{D}$, and select the better one under the budget.
8: **return** $\{b_1, b_2, \ldots, b_k\}$

The optimized perturbation $\delta^\star$ is not interpreted as the realized quantization error of a particular discrete bit-width assignment, but as a geometry-guided perturbation template. CET aggregates this template at the layer level to obtain a layer-wise geometric perturbation allowance δ_i, computed by summing the squared perturbation magnitude of $\delta^\star$ over the parameters of layer i. This quantity serves as a relative tolerance indicator under the local spectral geometry: a larger δ_i suggests that the corresponding layer is more compatible with geometry-favorable perturbations and can tolerate a more aggressive precision reduction in the subsequent bit-mapping stage.

Under standard high-resolution quantization theory [11], the distortion of a uniform quantizer scales approximately as $D \propto 2^{-2b}$. Motivated by this relationship [14, 34], CET maps the layer-wise geometric perturbation allowance to a continuous precision estimate and then discretizes it as

$$\tilde{b}_i = C - \frac{1}{2}\log_2(\delta_i), \qquad b_i = \text{clip}\Big(\Big\lfloor \tilde{b}_i \Big\rceil,\ b_{\min}, b_{\max}\Big), \tag{10}$$

where C is a global calibration term chosen such that the resulting bit-width assignment satisfies the target compression budget, and

Table 1: Comparison with existing uniform and mixed-precision quantization methods on CNNs and LLMs. MP denotes mixed precision. For CNNs, Metric denotes ImageNet Top-1; for LLMs, Metric denotes 5-shot MMLU average. w-ratio and a-ratio represent the weight and activation compression ratios, respectively. Red values indicate performance drop.

Model	Method	Metric/Full↑	w-bit	a-bit	w-ratio	a-ratio	Metric/Quant↑	Drop
ResNet-50	HAQ	76.15	MP	32	×10.57	×1.00	75.30	-0.85
	OBQ [10]	76.13	3	32	×10.66	×1.00	75.24	-0.89
	GPTQ [9]	76.13	3	32	×10.66	×1.00	74.87	-1.26
	CET (ours)	76.12	$2_{\mathbf{MP}}$	32	**×12.74**	×1.00	76.09	**-0.03**
	HAWQ-v2 [7]	77.39	$2_{\mathbf{MP}}$	$4_{\mathbf{MP}}$	×12.24	×8.00	75.76	-1.63
	PTMQ [26]	76.80	$4_{\mathbf{MP}}$	$4_{\mathbf{MP}}$	×8.00	×8.00	73.93	-2.87
	CET (ours)	76.12	$2_{\mathbf{MP}}$	$4_{\mathbf{MP}}$	**×13.53**	×8.00	75.13	**-0.99**
MobileNet-V2	HAQ [24]	71.87	MP	32	×14.07	×1.00	66.75	-5.12
	CET (ours)	71.83	$2_{\mathbf{MP}}$	32	**×14.99**	×1.00	**70.14**	**-1.69**
	MCKP [3]	71.88	$2_{\mathbf{MP}}$	8	×13.99	×4.00	68.52	-3.36
	PTMQ [26]	72.40	$4_{\mathbf{MP}}$	8	×8.00	×4.00	64.94	-7.46
	CET (ours)	71.83	$2_{\mathbf{MP}}$	8	**×14.99**	×4.00	**69.66**	**-2.17**
LLaMA2-7B	GPTQ [9]	45.70	4	8	×4.00	×2.00	43.80	-1.90
	QuIP [2]	45.70	4	8	×4.00	×2.00	42.94	-2.76
	CMPQ [4]	45.70	$4_{\mathbf{MP}}$	8	×4.00	×2.00	44.20	-1.50
	CET (ours)	45.70	$4_{\mathbf{MP}}$	8	**×4.21**	×2.00	**44.91**	**-0.79**
LLaMA3-8B	GPTQ [9]	64.80	4	8	×4.00	×2.00	63.02	-1.78
	AWQ [16]	64.80	4	8	×4.00	×2.00	63.20	-1.60
	DWR [29]	64.80	$4_{\mathbf{MP}}$	8	×4.00	×2.00	63.74	-1.06
	CET (ours)	64.80	$4_{\mathbf{MP}}$	8	**×4.21**	×2.00	**64.89**	**+0.09**

$b_{\min}, b_{\max}$ define the admissible precision range. In practice, layers with vanishingly small aggregated perturbation are assigned the maximum admissible precision.

Finally, CET refines the candidate precisions on the calibration set. Since Eq. 10 produces continuous estimates while practical deployment requires discrete bit-widths, CET performs a lightweight two-candidate refinement. For each layer, CET evaluates the rounded geometry-guided bit-width and one adjacent feasible discrete bit-width, and selects the better one according to the quantized loss under the global budget. In this way, the geometry-guided allocation determines the primary mixed-precision structure, while refinement provides only the minimal discrete correction required for deployment. To address the challenge of deploying mixed-precision quantization in practice, the heterogeneous precision configuration generated by CET can be executed by our custom mixed-precision inference engine, the practical implementation of which will be described in the experimental section.

4 Experiments

We organize the experiments around 3 research questions (RQ):

- **RQ1: How does CET perform across diverse model/tasks?**
- **RQ2: How does CET benefit from directional geometry?**
- **RQ3: How does CET perform in practical deployment?**

4.1 RQ1: Main Results Across Model Families

We first evaluate CET across vision, language understanding, large language models, and generative models to examine how well the same geometry-guided mixed-precision principle generalizes across heterogeneous architectures. Overall, CET consistently achieves strong compression–accuracy trade-offs, with the most pronounced gains appearing in mixed-precision and ultra-low-bit regimes, where quantization perturbations become larger and more directionally non-uniform. Unless otherwise stated, all methods are evaluated without retraining, using the same pretrained checkpoints and the same calibration setting for each task.

4.1.1 ***CNNs (Image Classification).*** Table 1 reports results on ResNet-50 [12], and MobileNet-V2 [20], covering uniform, Hessian-based, and recent mixed-precision baselines. CET performs strongly across all evaluated CNNs, with the clearest advantage appearing under aggressive low-bit settings.

On ResNet-50, CET outperforms HAWQ-v2 in the ultra-low-bit mixed setting of 2-bit weights and 4-bit activations, reducing the accuracy drop from 1.63% to 0.99% while achieving a higher weight compression ratio (13.53× vs. 12.24×). CET also compares favorably with search-based or weight-only baselines such as HAQ, OBQ, and GPTQ, achieving smaller degradation under stronger compression.

CET remains competitive at even lower precision. On MobileNet-V2, CET achieves 14.99× compression with only a 1.69% accuracy drop, substantially outperforming PTMQ and MCKP. These results suggest that CET is effective not only on standard CNN backbones, but also on lightweight architectures where quantization errors are typically harder to control. Figure 2 further shows that CET assigns highly non-uniform bit-widths across layers, consistent with its geometry-guided allocation principle.

4.1.2 ***Transformers (Natural Language Understanding).*** We next evaluate CET on transformer-based language understanding

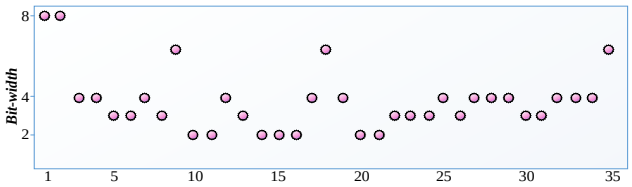

Figure 2: Bit-width assignments in ResNet-34.

Table 2: BERT-base performance on SQuAD1.1. We report performance drop after quantization.

Method	SQuAD (F1 / EM)	SWAG	w-ratio.	a-ratio.
FP32	88.42 / 80.89	79.11	×1	×1
ACIQ [1]	-4.28 / -3.55	-2.59	×6.07	×1
Zhang et al. [33]	-0.26 / -0.47	-0.81	×1.58	×1
CET (ours)	**-0.26 / -0.32**	**-0.07**	**×7.71**	**×2**
DirectQ	-11.59 / -15.49	–	×7.99	×4
Q-BERT [22]	-0.33 / -0.59	-1.21	×7.99	×4
ZeroQuant [28]	-0.31 / -0.50	-	×7.99	×4
CET (ours)	**-0.27 / -0.34**	**-0.99**	×7.99	×4

tasks, including SQuAD1.1 [19] and SWAG [30]. As shown in Table 2, CET consistently preserves performance close to the full-precision model while achieving substantially higher compression.

Under 4-bit weight quantization, CET incurs only a 0.26 drop in F1 on SQuAD1.1 and a 0.07 drop in SWAG accuracy, while outperforming ACIQ and Zhang et al. in compression ratio. Under 4-bit weight-and-activation quantization, CET still maintains only a 0.27 / 0.34 drop on SQuAD1.1 (F1 / EM) and a 0.99 drop on SWAG, substantially outperforming DirectQ and Q-BERT under the same compression ratio. These results indicate that CET's geometry-guided allocation is not limited to convolutional layers, but also transfers effectively to transformer architectures with highly non-uniform quantization behavior.

Table 3: Comparison results of commonsense reasoning under weighted quantization only in LLaMA-2 7B.

Method	Bits	Wikitext-2	ARC-c	ARC-e	HellaSwag	PIQA
LLaMA-2 7B	16	**5.12**	**43.3**	**76.3**	**57.1**	78.1
GPTQ	2_{MP}	50.75	20.9	34.9	30.5	57.2
AQLM	2_{MP}	6.29	34.9	66.5	50.88	74.92
QuIP# [23]	2_{MP}	6.19	34.6	64.6	51.91	75.1
CET (ours)	2_{MP}	**6.12**	**35.4**	**64.7**	**52.2**	**75.3**
GPTQ [9]	4_{MP}	5.49	36.8	66.2	55.4	76.6
AQLM [8]	4_{MP}	5.21	41.0	70.2	56.0	78.2
RSAVQ [27]	4_{MP}	5.22	42.0	74.7	56.3	76.9
VPTQ [18]	4_{MP}	5.26	39.7	69.0	56.0	78.1
CET (ours)	4_{MP}	**5.17**	**42.9**	**75.3**	**56.5**	**78.4**

4.1.3 ***LLM Quantization (Commonsense).*** We next evaluate CET on large language models to examine whether the proposed geometry-guided mixed-precision principle remains effective in autoregressive settings. Table 1 reports MMLU results for LLaMA2-7B and LLaMA3-8B. CET shows clear advantages over existing low-bit baselines on LLaMA2-7B and remains close to full-precision performance on LLaMA3-8B. This suggests that CET scales well to large autoregressive models under quantization.

To further probe this behavior beyond MMLU, we evaluate LLaMA-2 7B on WikiText-2 perplexity and zero-shot common-sense reasoning benchmarks. As shown in Table 3, CET consistently achieves the best overall performance under both 2_mix and 4_mix settings. Under 2_mix quantization, CET attains the lowest perplexity and the best accuracy on all downstream tasks among quantized methods. Under 4_mix quantization, CET remains highly competitive and achieves the strongest overall zero-shot performance. Together, these results indicate that CET preserves not only broad reasoning ability, but also language modeling quality and common-sense generalization under mixed-precision quantization.

Table 4: Quantization results on CIFAR-10 (32×32) with the pretrained DDIM sampler using 100 denoising time steps.

Method	w-ratio	a-ratio	GBops	FID↓	IS↑
Full Precision	×8	×4	399	4.26	9.03
PTQ4DM [21]	×8	×4	399	5.31	**9.24**
Q-Diffusion [15]	×8	×4	399	4.93	9.12
CET (Ours)	× 8	× 4	399	**4.88**	9.17

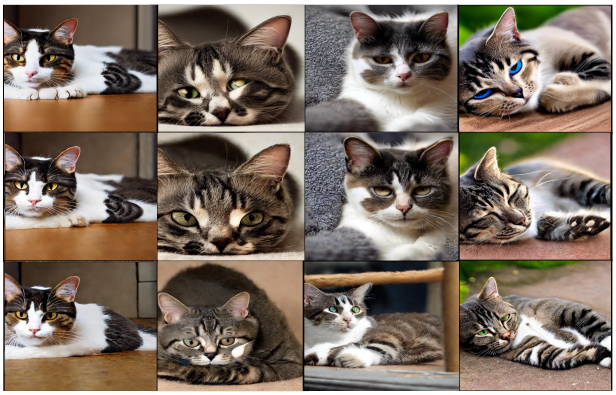

Figure 3: At FP32 (First row), W8A8 (second row) and W4A8 (third row) setting, using CET, stable diffusion 512 × 512 text-guided image synthesis results, the prompt is 'A tabby cat is lying down comfortably'.

4.1.4 ***Diffusion and Text-to-Image Generation.*** We further evaluate CET on generative models. On CIFAR-10 diffusion with a pretrained DDIM sampler, Table 4 shows that CET achieves the best FID among quantized baselines while maintaining competitive IS, indicating that the proposed geometry-guided allocation remains effective even when perturbations accumulate over iterative denoising steps. We also evaluate CET on Stable Diffusion v1.5 at 512×512 resolution. As shown in Figure 3, the generated images under W8A8 remain visually close to the FP32 baseline,

and the more aggressive W4A8 setting still preserves the main semantics and object structure specified by the prompt. These qualitative results are consistent with the diffusion results above and suggest that CET can preserve both structure and semantics under mixed-precision quantization in cross-modal generative settings.

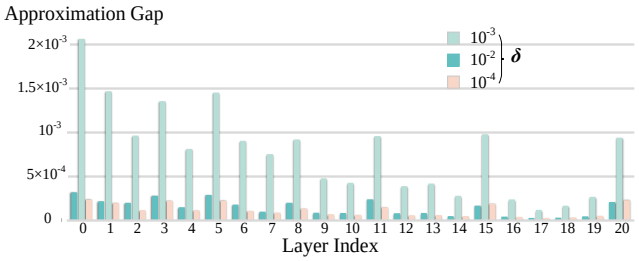

Figure 4: Layer-wise gap between true and quadratic loss under different perturbation magnitudes.

4.2 RQ2: Ablations

We next examine whether CET's performance gains are indeed supported by the proposed geometric mechanism. Specifically, we validate three aspects of the method:

(1) whether the local quadratic surrogate provides a reasonable approximation to quantization-induced loss variation, (2) whether restricting perturbations to the geometry-favorable subspace is beneficial in practice, and (3) whether CET improves over scalar sensitivity-based allocation and other variants.

4.2.1 ***Validity of the Local Quadratic Surrogate.*** We first examine whether the local second-order approximation in Eq. 2 provides an accurate surrogate for quantization-induced loss variation. As shown in Figure 4, we measure the gap between the loss predicted by the quadratic model and the actual loss under different perturbation magnitudes. Using ResNet-18 as a representative case, we observe that for most layers the theoretical and empirical losses remain well aligned when the perturbation scale lies in the range $[10^{-4}, 10^{-2}]$. This result supports the use of the local quadratic surrogate within a sufficiently local regime. The figure also shows that the approximation quality varies across layers and perturbation scales, indicating heterogeneous local sensitivity across model.

4.2.2 ***Effect of the Geometry-Favorable Subspace.*** We study the effect of the geometric subspace approximation used in CET. Since computing the full Hessian spectrum is infeasible for large models, CET uses a Lanczos-based spectral routine to retain a truncated set of leading positive-curvature eigenpairs to approximate the steep subspace V_{short}, while treating the remaining directions as the geometry-favorable subspace V_{long}. In practice, we do not form the full Hessian matrix explicitly; instead, we apply Lanczos to Hessian-vector products of the empirical calibration loss with respect to the full model parameters, and retain the leading positive-curvature eigenpairs.

We vary the number of retained eigenpairs (50, 100, 200, and 500). In Figure 5, increasing the retained rank consistently improves performance, indicating that a more accurate approximation of the steep local geometry yields better mixed-precision allocation. The

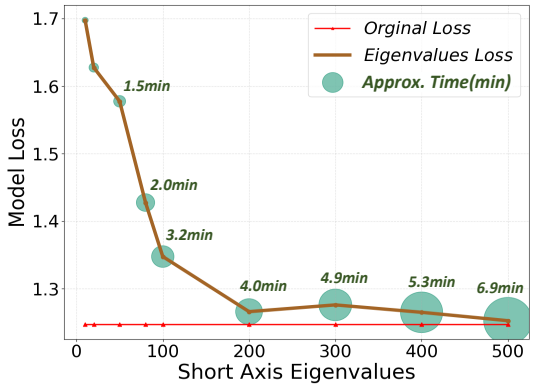

Figure 5: Effect of the rank used to approximate V_{short}.

gain saturates beyond 200, while the Lanczos-based Hessian overhead grows from 1.5 to 6.9 min. We therefore adopt 200 eigenpairs in practice as a favorable trade-off between fidelity and efficiency.

Table 5: Comparison with scalar sensitivity baselines and CET variants under the same weight compression ratio.

Variant	ImageNet (×13.5)	SWAG (×8)
Scalar sensitivity baselines		
Top-λ	72.18	76.08
Trace-based	73.89	75.27
Diag-Hessian	71.12	69.99
CET variants		
Random direction	61.41	40.03
CET w/o Subspace	67.20	61.19
CET w/o Solver	74.72	76.41
CET w/o Refinement	75.94	78.45
CET (Full)	**76.13**	**79.04**

4.2.3 ***Comparison with Scalar Sensitivity and CET Variants.*** We further compare CET with three groups of alternatives.

(1) Scalar sensitivity baselines. We compare CET with Top-λ, Trace-based allocation, and Diag-Hessian. These methods all use second-order information, but only as layer-wise scalar sensitivity.

(2) CET variants. We compare the full method with *Random direction*, which replaces the geometry-guided perturbation with a random one under the same perturbation budget, *CET w/o Subspace*, which removes the geometry-favorable subspace, *CET w/o Solver*, which replaces the solver in Eq. 9 with a normalized projection onto V_{long} while keeping the subsequent stages unchanged, and *CET w/o Refinement*, which removes the two-candidate correction.

Table 5 shows four trends. First, CET consistently outperforms scalar sensitivity baselines on both ImageNet and SWAG, indicating that its gain comes from directional geometric modeling rather than second-order layer ranking alone. Second, *Random direction* performs dramatically worse, confirming that perturbation direction is a real factor in quantization degradation. Third, *CET w/o Solver* still remains stronger than scalar sensitivity baselines, showing that

the geometry-favorable subspace already provides the dominant allocation signal. Finally, removing either the geometry-favorable subspace or the final refinement degrades performance, with the larger drop from removing the subspace indicating that avoiding steep directions is the main source of improvement, while the solver and refinement mainly provide additional gains.

(3) Calibration Set Size. Figure 6(right) shows the effect of calibration set size on WikiText-2 perplexity under 4_mix quantization. For both LLaMA-2 7B and 13B, performance improves as more calibration data are used, but the gain quickly saturates beyond moderate sample sizes. This indicates that CET is not overly sensitive to calibration set size in practice.

4.3 RQ3: Efficiency and Deployment

We finally examine whether CET is efficient in practice and whether the heterogeneous precision assignments produced by CET can be executed effectively at inference time. To answer this question, we analyze both the optimization cost of CET itself and the practical deployment of its mixed-precision outputs.

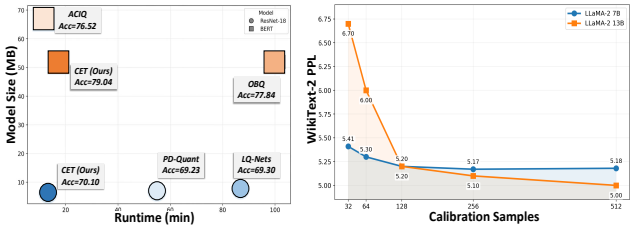

Figure 6: Left: Runtime–Performance comparison across representative methods. Right: sensitivity to calibration set size under 4_mix quantization on LLaMA-2 7B and LLaMA-2 13B.

4.3.1 ***Hessian Optimization Efficiency.*** CET is computationally efficient because it avoids both full fine-tuning and expensive reinforcement-learning-based bit-width search. In practice, the dominant cost comes from approximating the Hessian spectrum via Lanczos iterations, while the remaining stages—including subspace construction, layer-wise perturbation aggregation, and two-candidate refinement—are lightweight. As shown in the left panel of Figure 6, CET achieves a favorable trade-off between runtime and final performance. On both ResNet-18 and BERT, CET remains close to the desirable region of lower runtime and smaller model size, while also achieving the best quantized accuracy among the compared methods. This indicates that CET is not only effective, but also efficient in practice.

4.3.2 ***Inference Engine for Deployment.*** A practical challenge of mixed-precision quantization is that standard inference stacks are typically designed for uniform precision and thus cannot directly execute arbitrary layer-wise bit-width assignments. To make CET's heterogeneous allocation deployable in practice, we implement a custom mixed-precision inference backend that serves as a flexible execution layer connecting CET's geometry-guided bit allocation with the underlying runtime system. Given the bit-width configuration produced by CET, the backend parses the allocation, loads the corresponding quantized weights, and dispatches each layer to the matched precision-specific kernel at runtime, thereby providing an end-to-end path from allocation to execution without modifying the allocation itself.

At the kernel level, we adopt a bit-decomposition strategy similar to ABQ-LLM [31], which enables support for multiple precision modes (e.g., W2A8, W4A8, and W8A8) on NVIDIA Binary Tensor Core instructions. This design not only makes CET-generated heterogeneous precision plans practically runnable, but also improves deployment adaptability across different model scales, layer-wise precision patterns, and backend execution settings. **Additional implementation details and a schematic illustration are provided in the supplementary material**.

Table 6: Inference latency (ms) and memory usage (GB) of CET kernels on LLaMA-7B and LLaMA-30B.

Model	Method	Output Length 128		Output Length 512	
		Latency	Memory	Latency	Memory
LLaMA-7B	FP16	1490.50	13.47	6090.97	13.66
	CUTLASS (8-bit)	849.25	7.39	3764.22	7.59
	CUTLASS (4-bit)	657.33	4.25	2864.91	4.45
	CET-Kernel (2/4/8-bit)	655.84	4.13	**2801.33**	4.49
	CET-Kernel (2/4-bit)	**640.63**	**3.89**	2866.43	**4.00**
LLaMA-30B	FP16	3843.59	65.53	15241.36	66.11
	CUTLASS (8-bit)	3534.11	33.42	12993.34	33.21
	CUTLASS (4-bit)	2005.51	17.21	8111.17	17.82
	CET-Kernel (2/4/8-bit)	1979.05	16.06	8000.09	16.34
	CET-Kernel (2/4-bit)	**1899.61**	**12.90**	**7816.67**	**13.08**

4.3.3 ***Deployment Results.*** We evaluate the practical efficiency of the CET inference backend in terms of latency and memory usage on LLaMA-7B and LLaMA-30B. Table 6 reports inference latency and memory usage on LLaMA-7B and LLaMA-30B. Compared with FP16 and uniform CUTLASS baselines, the CET kernels consistently achieve lower memory usage while maintaining competitive or lower latency under heterogeneous settings such as 2/4/8-bit and 2/4-bit. For example, on LLaMA-30B with output length 512, the CET 2/4-bit kernel reduces memory usage from 17.82 GB (uniform 4-bit CUTLASS) to 13.08 GB, while also improving latency from 8111.17 ms to 7816.67 ms. These results show that CET's layer-wise mixed-precision assignments are not only effective as an allocation principle, but also executable and beneficial in practice.

For more details on **experimental setups, comparative results, and Hessian approximation**, please see the **Appendix**.

5 Conclusion

We presented CET, a geometry-guided framework for mixed-precision quantization. By modeling quantization as structured perturbations and analyzing their effect through a local second-order surrogate, CET uses Hessian spectral geometry to identify low-curvature directions and guide layer-wise precision allocation. Experiments across vision, language, and generative models show that CET achieves strong compression–accuracy trade-offs, particularly in mixed-precision and ultra-low-bit. We further show that the resulting heterogeneous precision assignments can be deployed efficiently in practice with a custom mixed-precision inference engine.

References

[1] Ron Banner, Yury Nahshan, Elad Hoffer, and Daniel Soudry. 2019. Post-training 4-bit quantization of convolution networks for rapid-deployment. arXiv:1810.05723 [cs.CV] https://arxiv.org/abs/1810.05723

[2] Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. 2024. QuIP: 2-Bit Quantization of Large Language Models With Guarantees. arXiv:2307.13304 [cs.LG] https://arxiv.org/abs/2307.13304

[3] Weihan Chen, Peisong Wang, and Jian Cheng. 2021. Towards mixed-precision quantization of neural networks via constrained optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5350–5359.

[4] Zihan Chen, Bike Xie, Jundong Li, and Cong Shen. 2025. Channel-Wise Mixed-Precision Quantization for Large Language Models. arXiv:2410.13056 [cs.CL] https://arxiv.org/abs/2410.13056

[5] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems* 36 (2023), 10088–10115.

[6] Zhen Dong, Zhewei Yao, Yaohui Cai, Daiyaan Arfeen, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. 2019. HAWQ-V2: Hessian Aware trace-Weighted Quantization of Neural Networks. arXiv:1911.03852 [cs.CV] https://arxiv.org/abs/1911.03852

[7] Zhen Dong, Zhewei Yao, Amir Gholami, Michael Mahoney, and Kurt Keutzer. 2019. HAWQ: Hessian AWare Quantization of Neural Networks with Mixed-Precision. arXiv:1905.03696 [cs.CV] https://arxiv.org/abs/1905.03696

[8] Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. 2024. Extreme Compression of Large Language Models via Additive Quantization. arXiv:2401.06118 [cs.LG] https://arxiv.org/abs/2401.06118

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

[10] Elias Frantar, Sidak Pal Singh, and Dan Alistarh. 2023. Optimal Brain Compression: A Framework for Accurate Post-Training Quantization and Pruning. arXiv:2208.11580 [cs.LG] https://arxiv.org/abs/2208.11580

[11] Robert M. Gray and David L. Neuhoff. 2002. Quantization. *IEEE transactions on information theory* 44, 6 (2002), 2325–2383.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[13] Zhiyong Huang, Xiao Han, Zhi Yu, Yunlan Zhao, Mingyang Hou, and Shengdong Hu. 2025. Hessian-based mixed-precision quantization with transition aware training for neural networks. *Neural Networks* 182 (2025), 106910. doi:10.1016/j.neunet.2024.106910

[14] Deokjae Lee and Hyun Oh Song. 2025. Q-Palette: Fractional-Bit Quantizers Toward Optimal Bit Allocation for Efficient LLM Deployment. *arXiv preprint arXiv:2509.20214* (2025).

[15] Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. 2023. Q-Diffusion: Quantizing Diffusion Models. arXiv:2302.04304 [cs.CV] https://arxiv.org/abs/2302.04304

[16] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration. In *Proceedings of Machine Learning and Systems*, P. Gibbons, G. Pekhimenko, and C. De Sa (Eds.), Vol. 6. 87–100. https://proceedings.mlsys.org/paper_files/paper/2024/file/42a452cbafa9dd64e9ba4aa95cc1ef21-Paper-Conference.pdf

[17] Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. 2025. QServe: W4A8KV4 Quantization and System Co-design for Efficient LLM Serving. arXiv:2405.04532 [cs.CL] https://arxiv.org/abs/2405.04532

[18] Yifei Liu, Jicheng Wen, Yang Wang, Shengyu Ye, Li Lyna Zhang, Ting Cao, Cheng Li, and Mao Yang. 2024. VPTQ: Extreme Low-bit Vector Post-Training Quantization for Large Language Models. arXiv:2409.17066 [cs.AI] https://arxiv.org/abs/2409.17066

[19] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* (2016).

[20] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2019. MobileNetV2: Inverted Residuals and Linear Bottlenecks. arXiv:1801.04381 [cs.CV] https://arxiv.org/abs/1801.04381

[21] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. 2023. Post-training Quantization on Diffusion Models. arXiv:2211.15736 [cs.CV] https://arxiv.org/abs/2211.15736

[22] Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. 2019. Q-BERT: Hessian Based Ultra Low Precision Quantization of BERT. arXiv:1909.05840 [cs.CL] https://arxiv.org/abs/1909.05840

[23] Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. 2024. QuIP#: Even Better LLM Quantization with Hadamard Incoherence and Lattice Codebooks. arXiv:2402.04396 [cs.LG] https://arxiv.org/abs/2402.04396

[24] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. 2019. Haq: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8612–8620.

[25] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2024. SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models. arXiv:2211.10438 [cs.CL] https://arxiv.org/abs/2211.10438

[26] Ke Xu, Zhongcheng Li, Shanshan Wang, and Xingyi Zhang. 2024. Ptmq: Post-training multi-bit quantization of neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 16193–16201.

[27] Zukang Xu, Xing Hu, Qiang Wu, and Dawei Yang. 2025. RSAVQ: Riemannian Sensitivity-Aware Vector Quantization for Large Language Models. arXiv:2510.01240 [cs.LG] https://arxiv.org/abs/2510.01240

[28] Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. 2022. ZeroQuant: Efficient and Affordable Post-Training Quantization for Large-Scale Transformers. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 27168–27183. https://proceedings.neurips.cc/paper_files/paper/2022/file/adf7fa39d65e2983d724ff7da57f00ac-Paper-Conference.pdf

[29] Hao Yu, Yang Zhou, Bohua Chen, Zelan Yang, Shen Li, Yong Li, and Jianxin Wu. 2025. Treasures in Discarded Weights for LLM Quantization. *Proceedings of the AAAI Conference on Artificial Intelligence* 39, 21 (Apr. 2025), 22218–22226. doi:10.1609/aaai.v39i21.34376

[30] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326* (2018).

[31] Chao Zeng, Songwei Liu, Yusheng Xie, Hong Liu, Xiaojian Wang, Miao Wei, Shu Yang, Fangmin Chen, and Xing Mei. 2025. ABQ-LLM: Arbitrary-Bit Quantized Inference Acceleration for Large Language Models. arXiv:2408.08554 [cs.LG] https://arxiv.org/abs/2408.08554

[32] Boyang Zhang, Daning Cheng, Yunquan Zhang, and Fangmin Liu. 2024. FP= xINT: A Low-Bit Series Expansion Algorithm for Post-Training Quantization. *arXiv preprint arXiv:2412.06865* (2024).

[33] Boyang Zhang, Suping Wu, Leyang Yang, Bin Wang, and Wenlong Lu. 2023. A Lightweight Grouped Low-rank Tensor Approximation Network for 3D Mesh Reconstruction From Videos. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 930–935.

[34] Chao Zhang, Li Wang, Samson Lasaulce, and Merouane Debbah. 2025. BAQ: Efficient Bit Allocation Quantization for Large Language Models. *arXiv preprint arXiv:2506.05664* (2025).