# Predicting Quantization Price for Selecting PTQ Configurations Before Deployment

**Junbin Qiu**[1]* **Jian Mu**[1]* **Weitong Zhang**[3] **Yao Shu**[1,2]†
[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]The Hong Kong University of Science and Technology
[3]Taylor Lab., Huawei Technologies Ltd.
{jqiu236,jmu817}@connect.hkust-gz.edu.cn
zhangweitong4@huawei.com, yaoshu@hkust-gz.edu.cn

## Abstract

Weight-space post-training quantization (PTQ) requires choices of finite formats, granularities, quantizer families, transformations, and bits before the output-distribution drift of the completed quantized model can be observed. Existing PTQ methods predict components of this degradation, including reconstruction error, Hessian sensitivity, transformation effects, and downstream loss, but these components are usually scored with the quantization geometry fixed or within separate configuration families. We formulate weight-space PTQ as pre-deployment configuration selection using priced layer-output error. Each admissible layer configuration is treated as an error generator that induces a layer-output error covariance $\mathbf{\Sigma}_l(\alpha_l)$ and has a deployment cost. The full-precision model prices this covariance using downstream curvature, $\hat{\rho}_l(\alpha_l) = \frac{1}{2}\operatorname{Tr}(\hat{\mathbf{H}}_l\hat{\mathbf{\Sigma}}_l(\alpha_l))$. The price follows from full-precision-to-quantized forward KL, whose first-order term cancels at the reference model. This formulation identifies reconstruction and diagonal scores as reduced proxies that omit price factors. Finite formats, codebooks, granularities, and equivalent transformations can then be compared through their induced covariances and deployment costs. A trace reduction yields a calibration-time price table and a budgeted price-guided selector, with fixed-geometry bit allocation as a special case of the broader configuration-selection problem. Code is available at https://github.com/Qjbtiger/PricePTQ.

## 1 Introduction

Weight-space PTQ requires configuration choices before their final effect on the model's output distribution can be observed. A deployment stack must choose the finite format, granularity, quantizer family, pre-quantization transformation, and bitwidth for each layer, but the resulting output-distribution drift can be evaluated only after these choices have been combined in a complete weight-quantized model. Building every candidate model is usually infeasible. Restricting the decision to mixed precision after fixing the geometry overlooks a central degree of freedom, since the configuration itself determines the injected layer-output error. The practical question is therefore how to rank admissible weight-space PTQ configurations before deployment trials, including how many bits to allocate. These undeployed choices must be scored in the same calibration pass.

Prior work already shows why this question cannot be answered by one local proxy alone. Reconstruction and second-order PTQ methods such as OBQ, OBC, GPTQ, AdaRound, BRECQ, QuIP, and QuIP# reduce or compensate local weight error [30, 20, 10, 11, 3, 40]. HAWQ-style methods,

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

HIGGS, and Q-Palette use Hessian, sensitivity, linear-response, or rate-distortion signals for precision or level allocation [8, 7, 29, 17]. SmoothQuant, AWQ, QuaRot, SpinQuant, OmniQuant, AffineQuant, and FlatQuant show that scaling, rotations, incoherence, outlier handling, and learned transforms change quantization behavior [46, 21, 1, 26, 34, 28, 39]. GuidedQuant and YAQA bring downstream loss or output-distribution information into quantization objectives [15, 42]. These results predict important pieces of PTQ degradation. The missing object is a common pre-deployment price that compares bits, granularities, codebooks, transformations, and hardware costs as configurations of the same decision. A detailed comparison with related methods is provided in Appx. A.

We use this price as the criterion for pre-deployment configuration selection. A layer configuration is an admissible weight-space PTQ choice supplied by the deployment stack, and it may change the candidate coordinate, specify a finite-format replacement, and carry a backend cost. Sec. 2 formulates configuration selection under a deployment budget, using full-precision-to-quantized forward KL as the target drift. Evaluating that drift requires a complete quantized model, so selection needs a calibration-time substitute based on the candidate-induced layer-output error covariance and the downstream curvature of the full-precision reference. The induced covariance and deployment cost connect these PTQ choices to the selector.

Sec. 3 derives this substitute from forward KL, without postulating a separate reconstruction score. At the full-precision reference, score cancellation removes the linear term, so the first retained local contribution is a quadratic price of candidate-induced layer-output error. The candidate supplies the covariance of that error, while the reference model supplies the output-side curvature that makes some error directions more costly than others. Familiar proxies follow by omitting factors from this price. Reconstruction omits the output curvature, and diagonal input scores additionally omit cross-channel input covariance. Such proxies can be useful reduced scores, but they preserve configuration rankings only when the omitted factors are effectively constant over the candidates being compared. The full price therefore serves as a configuration criterion that accounts for factors omitted by reconstruction.

The same price also explains how PTQ configuration families enter the selector. Finite formats, bits, granularities, vector codebooks, and palettes change the weight-perturbation moment being priced. Equivalent transformations change the coordinate in which that perturbation is generated and the input covariance it acts on, while leaving the reference-side output metric fixed. Under the isotropic trace reduction used for large models, each candidate can be scored by two candidate-side scalars and a cached curvature trace, then selected by a budgeted search over admissible costs. Fixed-geometry bit allocation is therefore recovered only after the transformation, quantizer family, and granularity have been held fixed. Before that point, PTQ is a broader configuration-selection problem over deployable choices, not an abstract continuous bit variable.

**Contributions.** The contribution centers on a price-prediction criterion for the configuration decision, using forward-KL and curvature signals to compare admissible PTQ choices across existing quantizer families. **First**, the paper identifies the object selected by weight-space PTQ as an admissible layer configuration that induces a layer-output error covariance and has a deployment cost. **Second**, it shows that full-precision-to-quantized forward KL assigns this covariance a quadratic quantization price, making reconstruction and diagonal scores interpretable as reduced prices. **Third**, it connects common PTQ families to the same price factors by showing that finite formats change perturbation moments, transformations change the candidate coordinate and input metric, and costs constrain deployability. **Fourth**, it turns these prices into a budgeted pre-deployment search rule that optimizes predicted quantization price directly, with mechanism evidence and scope boundaries organized around the same prediction chain.

## 2 Pre-Deployment Configuration Selection in Weight-Space PTQ

### 2.1 Reference Model, Candidate Coordinates, and Target Drift

Weight-space PTQ starts from a full-precision reference model and a calibration distribution $\mathcal{D}$, before any weight-quantized model has been deployed. Let the model have $L$ weight-bearing layers with reference weights $\mathbf{W}_1, \ldots, \mathbf{W}_L$. For a calibration input $\mathbf{x}$, let $\mathbf{x}_l$ be the reference input to layer $l$ and $\mathbf{z}_l = \mathbf{W}_l \mathbf{x}_l$ be its reference output. For each layer $l$, the deployment stack supplies an admissible set $\mathcal{A}_l$. A layer configuration $\alpha_l \in \mathcal{A}_l$ is the pre-deployment object that specifies bitwidth, granularity, quantizer family, hardware format, pre-quantization transformation, or a supported combination of

these choices. Candidate-dependent layer objects carry this local choice as an argument. When $\alpha_l$ changes coordinate before quantization, write the candidate coordinate as $\mathbf{x}_l(\alpha_l)$ and $\mathbf{W}_l(\alpha_l)$, which should satisfy the configuration invariance:

$$\mathbf{W}_l(\alpha_l)\mathbf{x}_l(\alpha_l) = \mathbf{W}_l\mathbf{x}_l \,, \tag{1}$$

where the ordinary weight quantization (without any coordinate change) is $\mathbf{x}_l(\alpha_l) = \mathbf{x}_l$, $\mathbf{W}_l(\alpha_l) = \mathbf{W}_l$. For a calibration batch, $\mathbf{X}_l(\alpha_l)$ collects the columns $\mathbf{x}_{l,i}(\alpha_l)$, and $\mathbf{X}_l$ denotes the reference-coordinate batch.

The same configuration also carries a deployment cost $\kappa_l(\alpha_l)$, and the deployment budget is $B$. A full configuration $\alpha = (\alpha_1, \ldots, \alpha_L)$ applies one layer choice to each layer. Applying it produces a weight-quantized model with output distribution $\pi_\alpha(\cdot \mid \mathbf{x})$, while the full-precision reference distribution is $\pi_0(\cdot \mid \mathbf{x})$. The quantity to be controlled is the final output-distribution drift:

$$\mathcal{J}(\alpha) \triangleq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[D_{\mathrm{KL}}\big(\pi_0(\cdot \mid \mathbf{x}) \,\|\, \pi_\alpha(\cdot \mid \mathbf{x})\big)\right] \,. \tag{2}$$

If this drift were known for every feasible configuration, PTQ would be the constrained selection problem

$$\min_{\alpha\in\mathcal{A}_1\times\cdots\times\mathcal{A}_L} \mathcal{J}(\alpha) \quad \text{s.t.} \quad \sum_{l=1}^{L} \kappa_l(\alpha_l) \le B \,. \tag{3}$$

This constrained program formalizes the deployment decision, but evaluating $\mathcal{J}(\alpha)$ requires a complete model incorporating the selected transformations, granularities, quantizer families, and bits. PTQ therefore needs a calibration-time rule for ranking configurations before this final model-level evaluation. We describe $\mathcal{A}_l$ through quantities available before that evaluation, including the candidate coordinate, finite-format replacement, and deployment cost.

### 2.2 Admissible PTQ Configuration Families and Costs

The admissible set $\mathcal{A}_l$ covers a broad range of configurations, but the price theory requires only quantities that each candidate can provide before deployment. We describe these candidates using the preceding coordinate convention. For a fixed layer, write $\alpha$ for its local choice and suppress the layer index. A candidate maps the reference calibration pair $(\mathbf{W}, \mathbf{X})$ to a floating-point coordinate $(\mathbf{W}(\alpha), \mathbf{X}(\alpha))$ that still satisfies $\mathbf{W}(\alpha)\mathbf{X}(\alpha) = \mathbf{W}\mathbf{X}$, and also supplies the finite-format replacement $\hat{\mathbf{W}}(\alpha)$ and the deployment cost $\kappa(\alpha)$. The coordinate equality means that pre-quantization transformations are equivalent before rounding, while $\hat{\mathbf{W}}(\alpha)$ is where finite-format error enters. The main families differ in how they choose the allowed values, how they change the coordinate, and what deployment cost they pay.

**Finite formats choose the allowed values.** A scalar group quantizer chooses a code set $\mathcal{C}_g$, a scale $s_g$, and a bitwidth $b_g$, then projects each group of weights onto those values:

$$\hat{\mathbf{W}}_g(\alpha) = s_g\, \Pi_{\mathcal{C}_g}(\mathbf{W}_g(\alpha)/s_g) \,, \qquad \delta_g = \frac{2r_g}{2^{b_g}-1} \,, \tag{4}$$

where $\Pi_{\mathcal{C}_g}$ is the projection onto the code set $\mathcal{C}_g$.

The goal is direct compression because smaller $b_g$ and coarser sharing reduce storage and kernel cost, but increase the local weight perturbation $\hat{\mathbf{W}}_g(\alpha) - \mathbf{W}_g(\alpha)$. Per-tensor, per-channel, group-wise, and block-wise quantizers differ by which weights share $s_g, r_g, b_g$ [8, 7]. Hardware floating-point and microscaling formats change the same choice through exponent sharing and admissible scales [35, 53]. Vector quantizers, lattice codes, trellises, and palettes replace the scalar code set by a block codebook. Their candidate-side moment summary is simply $\mathbf{V}_g(\alpha) \approx \mathbb{E}\left[\|\hat{\mathbf{W}}_g(\alpha) - \mathbf{W}_g(\alpha)\|_2^2\right]$ [17, 3, 40, 25, 41].

**Equivalent transformations change the coordinate before quantization.** Scaling, affine transforms, and rotations keep the floating-point layer unchanged, but make the finite-format problem easier:

$$\mathbf{X}(\alpha) = \mathbf{P}\mathbf{X}, \qquad \mathbf{W}(\alpha) = \mathbf{W}\mathbf{P}^{-1}, \qquad \mathbf{W}(\alpha)\mathbf{X}(\alpha) = \mathbf{W}\mathbf{X} \,. \tag{5}$$

These transformations keep the floating-point target fixed while moving quantization difficulty between the weight coordinate and the input coordinate [46, 21, 34, 28, 39]. Diagonal scaling $\mathbf{P} =$

$\mathrm{diag}(s)$ changes channel ranges and input magnitudes, which is why activation-aware scaling can protect important channels. Orthogonal rotations $\mathbf{P} = \mathbf{R},\ \mathbf{R}^\top\mathbf{R} = \mathbf{I}$ preserve Euclidean geometry while spreading outliers or improving incoherence before scalar, lattice, or trellis quantization [1, 26, 3, 40, 41]. Fast Hadamard-sign transforms $\mathbf{P} = \mathbf{HD}$ are the cheap deployable version when a full learned rotation is too expensive.

**Deployment costs define the deployable candidate set.** A candidate enters $\mathcal{A}$ only if the backend supports its format, metadata, and kernels. Along with the candidate coordinate and replacement weight, the selector therefore receives the deployment cost of the finite-format choice or transformation being scored:

$$\kappa(\alpha) = \kappa_{\text{bits}}(\alpha) + \kappa_{\text{meta}}(\alpha) + \kappa_{\text{kernel}}(\alpha)\ . \tag{6}$$

Bits and fractional-bit palettes mainly affect storage and arithmetic cost. Groups and codebooks add metadata or lookup overhead, and transformations may be folded offline or require online kernels [49, 17, 23]. The cost records which candidates are deployable under the budget. It does not measure accuracy degradation. Comparing these candidates before a complete quantized model is available also requires a common output-distribution price.

## 3 Priced Layer-Output Error for Configuration Selection

The formulation defines the constrained PTQ decision but does not yet provide a score for ranking configurations before the completed model is built. The target drift in (2) specifies the output-distribution quantity of interest, from which we seek a local calibration-time price for candidate layer configurations. The forward-KL expansion yields this price because its linear term vanishes at the full-precision reference.

### 3.1 Forward KL Induces the Layer-Output Error Price

The target $\mathcal{J}(\alpha)$ is the right fidelity measure, but evaluating it already requires a completed quantized model. The pre-deployment score must therefore be the price that the full-precision reference assigns locally to the error injected by a candidate. Using the candidate coordinate introduced in the setup, quantization produces $\hat{\mathbf{W}}_l(\alpha_l) = \mathbf{W}_l(\alpha_l) + \mathbf{\Delta}_l(\alpha_l)$ and injects $\mathbf{\Delta}_l(\alpha_l)\mathbf{x}_l(\alpha_l)$ at the reference layer output $\mathbf{z}_l = \mathbf{W}_l\mathbf{x}_l$. Set the input covariance $\mathbf{C}_l(\alpha_l) \triangleq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[\mathbf{x}_l(\alpha_l)\mathbf{x}_l(\alpha_l)^\top]$. For a calibration input $\mathbf{x}$, write $\mathcal{J}_{\mathbf{x},l}(\mathbf{z})$ for the per-sample forward-KL drift viewed as a function of the layer output $\mathbf{z}_l$, with the rest of the network held at the full-precision reference. The two layer statistics that price the perturbation are

$$\begin{aligned} \mathbf{H}_l &\triangleq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\nabla^2_{\mathbf{z}}\mathcal{J}_{\mathbf{x},l}(\mathbf{z})\big|_{\mathbf{z}=\mathbf{z}_l}\right]\ , \\ \mathbf{\Sigma}_l(\alpha_l) &\triangleq \mathbf{\Delta}_l(\alpha_l)\mathbf{C}_l(\alpha_l)\mathbf{\Delta}_l(\alpha_l)^\top\ . \end{aligned} \tag{7}$$

The first is the output-side curvature of the target drift, and the second is the candidate-induced layer-output error covariance. A PSD Gauss-Newton (GN) surrogate may replace $\mathbf{H}_l$ in implementation, but the formal object being priced is still the covariance generated by the candidate weight perturbation.

**Theorem 3.1** (Forward KL induces a quadratic quantization price)**.** *Assume the forward-KL objective is three-times differentiable on the weight segment from the full-precision reference to the candidate quantized model. At the reference, the first-order term vanishes. Keeping the block-diagonal second-order layer terms defined above, any candidate configuration $\alpha$ satisfies:*

$$\mathcal{J}(\alpha) = \rho(\alpha) + r(\alpha)\ , \qquad \rho(\alpha) \triangleq \sum_{l=1}^{L}\rho_l(\alpha_l)\ , \qquad \rho_l(\alpha_l) \triangleq \frac{1}{2}\,\mathrm{Tr}\left(\mathbf{H}_l\mathbf{\Sigma}_l(\alpha_l)\right)\ . \tag{8}$$

*Stack $\boldsymbol{\delta}_l \triangleq \mathrm{vec}(\mathbf{\Delta}_l(\alpha_l))$ into $\boldsymbol{\delta}$. Let $\mathbf{H}_\theta$ be the exact Hessian of $\mathcal{J}$ with respect to these stacked layer weights, and let $\tilde{\mathbf{H}}_\theta$ be the retained block-diagonal Hessian satisfying $\rho(\alpha) = \frac{1}{2}\boldsymbol{\delta}^\top\tilde{\mathbf{H}}_\theta\boldsymbol{\delta}$. If $M(\alpha)$ bounds the third derivative along the perturbation segment and $\rho(\alpha) > 0$, then*

$$|r(\alpha)| \le \eta(\alpha)\rho(\alpha)\ , \qquad \eta(\alpha) \triangleq \frac{\frac{1}{2}\|\mathbf{H}_\theta - \tilde{\mathbf{H}}_\theta\|_2\|\boldsymbol{\delta}\|_2^2 + \frac{M(\alpha)}{6}\|\boldsymbol{\delta}\|_2^3}{\rho(\alpha)}\ . \tag{9}$$

The proof is in Appx. B.1. The theorem assigns a price to the covariance induced by each configuration. The candidate supplies $\mathbf{\Sigma}_l(\alpha_l)$, and the reference model supplies the curvature $\mathbf{H}_l$ that makes some

output directions more costly than others. The relative diagnostic $\eta(\alpha)$ is used only to explain why the omitted term is negligible when the retained Hessian is accurate along the quantization direction and the perturbation remains local. In this regime, the selector optimizes the quadratic price itself, whose factors determine how PTQ configurations can be compared.

**Remark 3.2** (Reconstruction and diagonal scores are price reductions)**.** Fix a layer and suppress $l$ and the candidate argument. Let $\mathbf{X} = [\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}]$, $\mathbf{C} = N^{-1}\mathbf{X}\mathbf{X}^\top$, and $\mathbf{\Delta_W} = \hat{\mathbf{W}} - \mathbf{W}$. The full layer price $\rho = \frac{1}{2}\operatorname{Tr}(\mathbf{H}\mathbf{\Delta_W}\mathbf{C}\mathbf{\Delta_W}^\top)$ becomes the layer reconstruction score by the substitution $\mathbf{H} \mapsto \mathbf{I}$, giving $\rho_{\text{rec}} = \frac{1}{2}\operatorname{Tr}(\mathbf{\Delta_W}\mathbf{C}\mathbf{\Delta_W}^\top) = \frac{1}{2N}\|\mathbf{\Delta_W}\mathbf{X}\|_F^2$. A diagonal input-statistic score makes one further substitution, $\mathbf{C} \mapsto \operatorname{diag}(\mathbf{C})$, giving $\rho_{\text{diag}} = \frac{1}{2}\sum_j [\mathbf{C}]_{jj}\|[\mathbf{\Delta_W}]_{:,j}\|_2^2$. Thus reconstruction drops downstream output curvature, while diagonal scores also drop cross-channel input covariance. These reductions preserve candidate rankings only when the dropped factors are effectively constant over the candidate set. Otherwise reliable selection keeps the factors of the full price explicit.

### 3.2 From Candidate Families to Price Factors

The price theorem expresses each candidate's contribution through its expected layer-output error covariance. The selector therefore needs this covariance for every candidate available before deployment. The candidate families in Sec. 2.2 provide the quantities needed to determine it, including a candidate coordinate $(\mathbf{W}_l(\alpha_l), \mathbf{X}_l(\alpha_l))$, a quantized replacement $\hat{\mathbf{W}}_l(\alpha_l)$ or perturbation moment envelope $\mathbf{V}_l(\alpha_l)$, and a deployment cost $\kappa_l(\alpha_l)$. The reference-side curvature $\mathbf{H}_l$ is unchanged by these choices. Finite formats change the perturbation moment, equivalent transformations change the coordinate in which that perturbation is generated, and the cost enters the search constraint. These two configuration classes determine how candidate choices induce covariances. The scoring reduction simplifies computation and does not define an additional candidate family.

**Configuration class for finite formats, bits, and granularity.** This class fixes the finite values a weight may take. Bitwidth, per-tensor, per-channel or group-wise sharing, scalar and microscaling formats, vector formats, and codebooks all determine the perturbation moment created by the replacement weight. Before the completed model is built, the candidate can therefore provide the moment envelope $\mathbf{V}_l(\alpha_l)$ introduced in Sec. 2.2. Under the usual zero-mean independent-entry rounding envelope, with $\mathbf{h}_l$ and $\mathbf{c}_l$ collecting the diagonals of $\hat{\mathbf{H}}_l$ and $\mathbf{C}_l(\alpha_l)$, the expected price obeys:

$$\mathbb{E}[\hat{\rho}_l(\alpha_l)] \le \frac{1}{2}\,\mathbf{h}_l{}^\top \mathbf{V}_l(\alpha_l)\mathbf{c}_l\,. \tag{10}$$

Appx. B.2 derives the bound from the scalar rounding model. The display shows where the familiar bit law enters. For a per-layer uniform range $r_l$, $\delta_b = 2r_l/\mathsf{L}_b$ and $\mathbf{V}_l(b) = \delta_b^2\mathbf{1}\mathbf{1}^\top/12$, which gives

$$\mathbb{E}[\hat{\rho}_l(b)] \le \frac{r_l^2}{6\mathsf{L}_b^2}\operatorname{Tr}(\hat{\mathbf{H}}_l)\operatorname{Tr}(\mathbf{C}_l)\,, \qquad \mathsf{L}_b = 2^b - 1\,. \tag{11}$$

Thus the $4^{-b}$ decay is not a standalone allocation law. It appears only after the format has been converted into the perturbation moment being priced. Granularity changes which entries share a range or step size, while vector and codebook quantizers enter the same price expression by replacing scalar variances with empirical block perturbation moments.

**Configuration class for equivalent transformations.** This class contains pre-quantization transformations that preserve the floating-point layer while changing the coordinate in which finite-format error is generated. For an invertible transformation $\tilde{\mathbf{x}} = \mathbf{P}\mathbf{x}$ and $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{P}^{-1}$, the exact floating-point output is unchanged, but the quantized residual $\tilde{\mathbf{\Delta}}_b(\mathbf{P}) \triangleq Q_b(\mathbf{W}\mathbf{P}^{-1}) - \mathbf{W}\mathbf{P}^{-1}$ is generated on the transformed weight and is priced against the transformed input covariance. An optimized transformation family $\mathcal{P}_l$ therefore contributes the candidate covariance through

$$\begin{aligned} \hat{\mathbf{\Sigma}}_l(\mathbf{P}, b) &= \mathbb{E}\left[\tilde{\mathbf{\Delta}}_b(\mathbf{P})(\mathbf{P}\mathbf{C}\mathbf{P}^\top)\tilde{\mathbf{\Delta}}_b(\mathbf{P})^\top\right]\,, \\ \mathbf{P}^\star &\in \operatorname*{arg\,min}_{\mathbf{P}\in\mathcal{P}_l} \frac{1}{2}\operatorname{Tr}\left(\hat{\mathbf{H}}_l\hat{\mathbf{\Sigma}}_l(\mathbf{P}, b)\right)\,. \end{aligned} \tag{12}$$

This expression is the transformation counterpart of the finite-format moment bound. The matrix $\hat{\mathbf{H}}_l$ provides fixed reference-side curvature, while the transformation trades off the candidate-side factors

$\mathbf{PCP}^\top$ and $\tilde{\mathbf{\Delta}}_b(\mathbf{P})$. The affine case isolates the input-metric effect before the finite-format residual is reintroduced.

**Theorem 3.3** (Affine preprocessing has a whitening optimum)**.** *Let $\mathbf{C} \succ 0$ with dimension $n$ and set $\lambda_g = (\det \mathbf{C})^{1/n}$. Among determinant-preserving invertible input transformations,*

$$\min_{\mathbf{P}\in \mathrm{GL}(n):\, |\det \mathbf{P}|=1} \lambda_{\max}(\mathbf{PCP}^\top) = \lambda_g \ . \tag{13}$$

*Equality holds if and only if $\mathbf{PCP}^\top = \lambda_g \mathbf{I}$. Equivalently, if $\mathbf{C} = \mathbf{U}\Lambda\mathbf{U}^\top$, then $\mathbf{P} = \lambda_g^{1/2}\mathbf{Q}\Lambda^{-1/2}\mathbf{U}^\top$ for some $\mathbf{Q} \in \mathrm{O}(n)$.*

The theorem says only what a fully optimized affine transform can do to the input metric. Under a determinant normalization, the smallest possible worst-direction variance is achieved by whitening. If the quantizer residual were fixed and isotropic, this would reduce the candidate-side covariance to a scalar multiple of the identity and leave only the output-curvature trace. In weight PTQ, however, $\mathbf{P}$ also changes $\mathbf{WP}^{-1}$ and therefore the finite-format residual. Practical transformations consequently optimize a restricted version of (12), balancing input-metric flattening against transformed-weight distortion. Appx. B.4 proves the whitening claim, and Appx. B.3 records the covariance identity used by the transformed price.

Restricted transformation families inherit the same price object with less freedom in $\mathbf{P}$. Diagonal scaling $\mathbf{S} = \mathrm{diag}(s)$ gives

$$\hat{\rho}_{\mathbf{S},b} = \frac{1}{2}\,\mathrm{Tr}\left(\hat{\mathbf{H}}\,\mathbb{E}[\tilde{\mathbf{\Delta}}_b(\mathbf{S})(\mathbf{SCS})\tilde{\mathbf{\Delta}}_b(\mathbf{S})^\top]\right) \ , \quad \tilde{\mathbf{\Delta}}_b(\mathbf{S}) = Q_b(\mathbf{WS}^{-1}) - \mathbf{WS}^{-1} \ . \tag{14}$$

Scaling can lower price either by reducing the residual of $\mathbf{WS}^{-1}$ or by reshaping variances in $\mathbf{SCS}$, but it cannot remove normalized input correlations (Appx. B.7). Orthogonal rotations obey the same transformed-price form,

$$\hat{\rho}_{\mathbf{R},b} = \frac{1}{2}\,\mathrm{Tr}\left(\hat{\mathbf{H}}\,\mathbb{E}[\tilde{\mathbf{\Delta}}_b(\mathbf{R})(\mathbf{RCR}^\top)\tilde{\mathbf{\Delta}}_b(\mathbf{R})^\top]\right) \ , \quad \tilde{\mathbf{\Delta}}_b(\mathbf{R}) = Q_b(\mathbf{WR}^\top) - \mathbf{WR}^\top \ . \tag{15}$$

Because $\mathbf{RCR}^\top$ preserves the eigenvalues of $\mathbf{C}$, rotations help through the finite-format residual, range, or coherence rather than through input whitening. Appx. B.8 gives the corresponding scalar-quantizer distortion bound. Transformations therefore change the candidate covariance, but the output metric that prices that covariance remains the reference-side $\mathbf{H}_l$.

**Scoring reduction for isotropic trace price.** The isotropic trace reduction is a computational simplification, not a configuration class. It provides a low-cost statistic when a candidate family has made both the input metric and the perturbation moment nearly isotropic. The covariance-level formulas still define price, while near isotropy allows the selector to avoid forming full matrices. If $\mathbf{C} = \mathbf{U}\Lambda\mathbf{U}^\top$ and $\mathbf{P} = \sigma\mathbf{Q}\Lambda^{-1/2}\mathbf{U}^\top$, then $\mathbf{PCP}^\top = \sigma^2\mathbf{I}$. When the candidate perturbation moment is $\mathbf{R}_l(\alpha_l) \triangleq \mathbb{E}[\mathbf{\Delta}_l(\alpha_l)\mathbf{\Delta}_l(\alpha_l)^\top] = \omega_l(\alpha_l)\mathbf{I} + \mathbf{A}_l(\alpha_l)$, the price reduces to

$$\rho_l(\alpha_l) = \frac{1}{2}\sigma_l^2(\alpha_l)\omega_l(\alpha_l)\,\mathrm{Tr}(\mathbf{H}_l) + \frac{1}{2}\sigma_l^2(\alpha_l)\,\mathrm{Tr}\left(\mathbf{H}_l\mathbf{A}_l(\alpha_l)\right) \ . \tag{16}$$

If the anisotropic residual $\mathbf{A}_l$ is small, the candidate is ranked by the single coefficient $\sigma_l^2(\alpha_l)\omega_l(\alpha_l)$ times the cached reference-side trace $\mathrm{Tr}(\mathbf{H}_l)$. The reduction does not replace $\mathbf{H}_l$ by $\mathbf{I}$. It keeps the output metric through its trace only after the candidate-side covariance has been made nearly isotropic. Appx. B.6 gives the residual bound, and Appx. B.9 records the diagnostics for regimes where diagonal curvature, off-diagonal input covariance, or imatrix-style reductions drop factors that are not constant over the candidate set. Once the trace reduction is valid, each candidate can be scored by two candidate-side scalars and one reference-side curvature trace.

### 3.3 Budgeted Price-Guided Configuration Search

The trace reduction provides a computable score for each candidate, while $\kappa_l(\alpha_l)$ determines whether the candidate fits the deployment budget. This score is part of the search procedure and does not define an additional configuration class. For calibration samples $\{\mathbf{x}^{(i)}\}_{i=1}^N$, input dimension $n_l$, and output dimension $m_l$, the implementation estimates the candidate-side input scale and perturbation scale as

$$\hat{\sigma}_l^2(\alpha_l) \triangleq \frac{1}{n_l}\,\mathrm{Tr}\left(\frac{1}{N}\sum_{i=1}^N \mathbf{x}_{l,i}(\alpha_l)(\mathbf{x}_{l,i}(\alpha_l))^\top\right) \ , \qquad \hat{\omega}_l(\alpha_l) \triangleq \frac{1}{m_l}\,\mathrm{Tr}\left(\mathbf{\Delta}_l(\alpha_l)\mathbf{\Delta}_l(\alpha_l)^\top\right) \ . \tag{17}$$

When the candidate is represented by a moment envelope rather than an explicit replacement weight, the same normalized trace is taken from that envelope. The reference model supplies $\hat{\tau}_l \triangleq \mathrm{Tr}(\hat{\mathbf{H}}_l)$, estimated directly in large models by a PSD Gauss-Newton (GN) trace estimator such as Hutchinson [14] rather than by forming $\hat{\mathbf{H}}_l$. The implemented score is therefore

$$\hat{\rho}_l(\alpha_l) \leftarrow \frac{1}{2}\hat{\sigma}_l^2(\alpha_l)\hat{\omega}_l(\alpha_l)\hat{\tau}_l \ . \tag{18}$$

The candidate supplies $\hat{\sigma}_l^2(\alpha_l)$, $\hat{\omega}_l(\alpha_l)$, and $\kappa_l(\alpha_l)$, while the reference model supplies the cached curvature trace $\hat{\tau}_l$. Thus the expensive model-level trial is replaced by a calibration-time table of candidate prices and costs.

The remaining operation is the constrained selection itself. As in training-free NAS [37, 38], the selector scores many candidates before building their completed models. The ideal but unavailable target in (3) is replaced by the predicted configuration price $\hat{\rho}(\alpha) \triangleq \sum_{l=1}^{L} \hat{\rho}_l(\alpha_l)$, while the deployment constraint is left unchanged:

$$\alpha^\star \in \arg\min_{\alpha \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_L} \hat{\rho}(\alpha) \quad \text{s.t.} \quad \sum_{l=1}^{L} \kappa_l(\alpha_l) \le B \ . \tag{19}$$

Equation (19) is the pre-deployment replacement for exhaustive deployment trials. The set $\mathcal{A}_l$ supplies admissible choices, $\kappa_l(\alpha_l)$ supplies deployment cost, and the score in (18) supplies the objective being minimized. Alg. 1 gives the resulting selector.

**Algorithm 1** Price-Guided Configuration Search

**Require:** Reference model, calibration set, $\{(\mathcal{A}_l, \kappa_l)\}_{l=1}^{L}$, budget $B$.
**Ensure:** Selected configuration $\alpha^\star$.

```
Estimate and cache {τ̂_l = Tr(Ĥ_l)}_{l=1}^L once from the full-precision forward-KL curvature.
for l = 1, ..., L do
    for α_l ∈ A_l do
        Estimate candidate scalars σ̂_l^2(α_l) and ω̂_l(α_l), then score ρ̂_l(α_l) ← (1/2) σ̂_l^2(α_l) ω̂_l(α_l) τ̂_l.
    end for
end for
Select α* by solving the budgeted search problem in (19).
return α*.
```

## 4 Experiments

The price-guided selector in Alg. 1 is useful only if its calibration-time scores preserve the target decision before the quantized model is built. The experiments therefore test the prediction chain in the same order as the method. First, controlled perturbation studies ask whether the realized price in (8) and the implemented trace price in (18) track the full-precision-to-quantized KL drift (Sec. 4.1). Second, end-to-end weight-only PTQ experiments ask whether optimizing the predicted price reduces configuration-search cost while preserving perplexity and downstream task accuracy (Sec. 4.2).

### 4.1 Realized Price and Implemented-Trace-Price Align with KL Drift

The selector depends on the alignment between price and KL drift. If the realized layer-output price (8) and the implemented trace price (18) fail to track the final KL drift, then the price is not a valid surrogate for the target in (2). For each controlled candidate, we build the corresponding quantized perturbation, evaluate the full-precision-to-quantized KL drift on held-out calibration inputs, and compare it with the realized price and the implemented trace price. We compare these quantities on OPT-125M [54] and Qwen3-0.6B [48], across candidates that vary in transformation, bitwidth, and granularity. The transformation candidates include three types of whitening transformations (see Appx. B.5) and the no-transformation baseline. The detailed experimental setup is provided in Appx. D.

Fig. 1 shows that the realized price closely tracks the measured KL drift, with correlations $\rho = 0.9470$ on OPT-125M and $\rho = 0.9249$ on Qwen3-0.6B. This is the expected behavior of the price theorem in

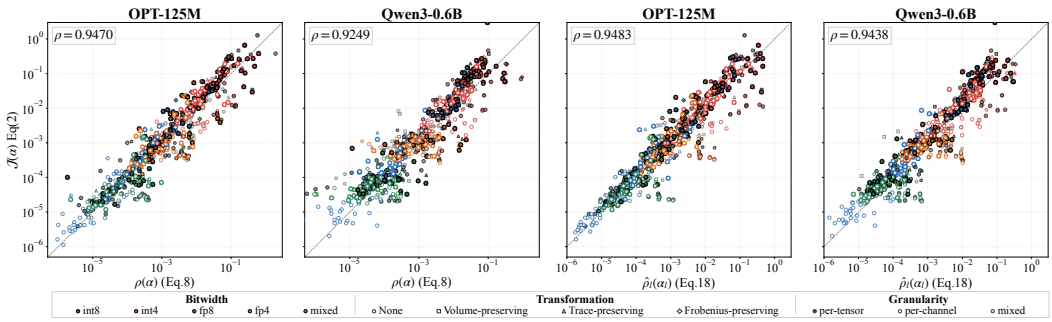


Figure 1: Realized quantization price (8) (left two panels) and implemented trace price (18) (right two panels) versus full-precision-to-quantized KL drift (2) under controlled perturbations. Each point is a candidate configuration, with marker styles indicating transformation constraints and colors indicating bitwidth or granularity choices. The reported $\rho$ summarizes the log-scale alignment.

Thm. 3.1. Although the controlled perturbation is applied to a single layer, we also report a two-layer price, shown by smaller markers in Fig. 1, which sums the price of the perturbed layer and a second layer with a different candidate perturbation. The two-layer price also tracks the KL drift, supporting the local linearized decomposition in Thm. 3.1 and consistent with the Linearity Theorem's claim that price is a local property of the quantization direction rather than a global property of the completed quantized model [29].

The implemented trace price provides a more stringent test because it omits the full covariance calculation and retains only the candidate-side scales $\hat{\sigma}_l^2(\alpha_l)$, $\hat{\omega}_l(\alpha_l)$, and the cached reference-side trace $\hat{\tau}_l$. Despite this reduction, the right two panels retain nearly the same alignment with KL drift, with $\rho = 0.9483$ on OPT-125M and $\rho = 0.9438$ on Qwen3-0.6B. In this controlled regime, it preserves the ordering induced by the realized quantization price. This supports the use of (18) as a low-cost pre-deployment quantization-price statistic, distinct from a reconstruction score.

### 4.2 Price-Guided Configuration Selection

The KL comparisons establish price as a meaningful ranking signal. We therefore test whether this signal improves discrete PTQ decisions before deployment. We evaluate these decisions on Llama-3.2-1B, Llama-3.2-3B, and Llama-3.1-8B across three representative configuration classes: bit allocation, transformation selection, and granularity allocation (see Appx. D for details).

Tab. 1 summarizes the results on Llama-3.2-1B. For bit allocation, each linear layer chooses a bitwidth from the admissible low-bit set $[2, 3, 4]$ under a target average of 3.0 bits, and we compare against AMQ [19] and HIGGS [29] allocation. AMQ achieves the lowest WikiText2 PPL in this run, but requires substantially more allocation time because its search builds and evaluates a quantized model for each candidate allocation. The price-guided allocation achieves the best average downstream task score among the three 3-bit methods. Compared with HIGGS, it achieves lower PPL and higher average task accuracy with less selection time. Although these results do not establish price as the best proxy across all metrics, they show that optimizing the predicted price can find a competitive allocation at substantially lower selection cost. Besides, our method has an over $1000\times$ speedup compared to AMQ, which is a significant advantage in practical deployment scenarios.

For transformation selection, the quantizer is held fixed while the selector chooses among admissible pre-quantization transformations, with CALM-CKA [52] and random choice as baselines. Furthermore, our method obtains an approximately $10.3\times$ speedup over CALM-CKA. For granularity allocation, the selector chooses the scale at which weights share quantization parameters, with fixed group-128 and random choice as baselines. On Llama-3.2-1B, the price-guided choice gives the best PPL and the best average task score in both settings. This indicates that pricing the induced layer-output error can select effective coordinates and granularities rather than treating these choices as backend details.

To test whether the same selection rule remains effective as model size increases, we apply the same protocol to Llama-3.2-3B and Llama-3.1-8B. The three comparisons are unchanged: bit allocation

Table 1: Price-guided configuration selection results on Llama-3.2-1B. The FP16 row is the unquantized reference. PPL is evaluated on WikiText2. Selection time measures the duration of the core configuration decision procedure and excludes the subsequent quantization and evaluation pass. The best and second-best results are highlighted in **bold** and underline, respectively.

| Method | Time (s) ↓ | PPL ↓ | BoolQ ↑ | TruthfulQA ↑ | PIQA ↑ | WinoGrande ↑ | WiC ↑ | Avg. ↑ |
|---|---|---|---|---|---|---|---|---|
| FP16 | - | 9.75 | 63.82 | 38.48 | 54.46 | 60.14 | 44.83 | 52.35 |
| **Bit Allocation** | | | | | | | | |
| HIGGS [29] | 813 | 12.50 | 58.23 | 40.24 | 49.67 | **56.91** | 47.81 | 50.57 |
| AMQ [19] | 11635 | **11.05** | 55.26 | 40.12 | **50.82** | 56.04 | **48.43** | 50.13 |
| Ours | **8.33** | 12.22 | **62.42** | **41.85** | 49.51 | 55.72 | 46.71 | **51.24** |
| **Transformation Selection** | | | | | | | | |
| CALM-CKA [52] | 703 | 12.80 | 50.92 | 37.59 | 49.51 | 56.12 | **51.88** | 49.20 |
| Random Selection | - | 22.68 | 48.00 | **39.53** | 49.75 | 54.56 | 49.84 | 48.34 |
| Ours | **67.96** | **11.53** | **53.64** | 38.67 | **49.78** | **57.30** | 49.69 | **49.82** |
| **Granularity Allocation** | | | | | | | | |
| Fixed Group-128 | - | 12.71 | **63.49** | 35.96 | 51.96 | 56.20 | 49.84 | 51.49 |
| Random Selection | - | 23313 | 43.57 | **48.64** | 3.10 | 50.62 | **51.15** | 39.41 |
| Ours | **29.83** | **11.39** | 62.54 | 36.90 | **52.99** | **58.01** | 49.37 | **51.96** |

compares with HIGGS and AMQ after fixing the GPTQ backend, transformation selection compares with CALM-CKA and random selection, and granularity allocation compares with fixed group-128 and random selection.

Table 2: Price-guided configuration selection results on Llama-3.2-3B. The FP16 row is the unquantized reference. PPL is evaluated on WikiText2. Selection time measures the duration of the core configuration decision procedure and excludes the subsequent quantization and evaluation pass. The best and second-best results are highlighted in **bold** and underline, respectively.

| Method | Time (s) ↓ | PPL ↓ | BoolQ ↑ | TruthfulQA ↑ | PIQA ↑ | WinoGrande ↑ | WiC ↑ | Avg. ↑ |
|---|---|---|---|---|---|---|---|---|
| FP16 | - | 7.81 | 74.04 | 39.28 | 72.25 | 68.43 | 50.31 | 60.86 |
| **Bit Allocation** | | | | | | | | |
| HIGGS [29] | 3417 | 13.18 | 51.10 | **40.39** | 18.55 | 55.64 | 49.69 | 43.08 |
| AMQ [19] | 16265 | **9.41** | 57.03 | 36.58 | **60.34** | **63.85** | 48.75 | 53.31 |
| Ours | **14.27** | 11.23 | **61.99** | 39.10 | 54.52 | 60.54 | **52.04** | **53.64** |
| **Transformation Selection** | | | | | | | | |
| CALM-CKA [52] | 970 | 8.81 | 71.90 | **40.12** | 67.79 | 64.48 | **48.59** | 58.58 |
| Random Selection | - | 10.68 | 67.45 | 38.95 | 63.98 | 63.27 | 48.54 | 56.44 |
| Ours | **159.47** | **8.58** | **72.26** | 39.75 | **70.18** | **65.43** | 47.18 | **58.96** |
| **Granularity Allocation** | | | | | | | | |
| Fixed Group-128 | - | 8.81 | 71.62 | 39.99 | 68.17 | 65.59 | 46.55 | **58.38** |
| Random Selection | - | 2504.78 | 44.33 | **47.33** | 6.93 | 50.38 | **49.27** | 39.65 |
| Ours | **73.34** | **8.46** | **72.11** | 37.98 | **68.50** | **65.67** | 46.39 | 58.13 |

Table 3: Price-guided configuration selection results on Llama-3.1-8B. The FP16 row is the unquantized reference. PPL is evaluated on WikiText2. Selection time measures the duration of the core configuration decision procedure and excludes the subsequent quantization and evaluation pass. The best and second-best results are highlighted in **bold** and underline, respectively.

| Method | Time (s) ↓ | PPL ↓ | BoolQ ↑ | TruthfulQA ↑ | PIQA ↑ | WinoGrande ↑ | WiC ↑ | Avg. ↑ |
|---|---|---|---|---|---|---|---|---|
| FP16 | - | 6.24 | 83.00 | 44.17 | 77.09 | 71.35 | 51.10 | 65.34 |
| **Bit Allocation** | | | | | | | | |
| HIGGS [29] | 5805 | 11.15 | 58.99 | 39.56 | 28.40 | 58.72 | 49.69 | 47.07 |
| AMQ [19] | 23337 | **9.31** | **63.73** | **41.70** | **55.88** | **62.90** | **50.00** | **54.84** |
| Ours | **31.91** | 10.87 | 60.83 | 38.74 | 36.67 | 60.46 | **50.00** | 49.34 |
| **Transformation Selection** | | | | | | | | |
| CALM-CKA [52] | 1957 | 6.97 | **81.38** | 42.19 | 76.61 | **70.72** | 50.00 | 64.18 |
| Random Selection | - | 24.34 | 65.06 | **42.66** | 34.69 | 61.19 | 48.64 | 50.45 |
| Ours | **326.23** | **6.86** | 80.43 | 41.80 | **77.48** | 70.56 | **50.94** | **64.24** |
| **Granularity Allocation** | | | | | | | | |
| Fixed Group-128 | - | 7.02 | 80.89 | 40.39 | 71.76 | 70.64 | **50.00** | 62.74 |
| Random Selection | - | 2326.04 | 45.28 | **48.73** | 8.94 | 50.59 | 49.32 | 40.57 |
| Ours | **165.83** | **6.75** | **81.47** | 41.28 | **74.97** | **71.11** | **50.00** | **63.77** |

Tabs. 2 and 3 show a consistent pattern for transformation and granularity choices, where the selected configuration changes how the candidate perturbation induces layer-output error. In transformation selection, the price-guided selector gives the best PPL and the best average task score on both models. On Llama-3.2-3B it improves over CALM-CKA from 8.81 to 8.58 PPL and from 58.58 to 58.96 average score. On Llama-3.1-8B it similarly improves PPL from 6.97 to 6.86 and gives a slightly higher average score, 64.24 versus 64.18, with a lower reported selection time than CALM-CKA. For granularity allocation, the price-guided choice gives the best PPL on both models and the best average score on Llama-3.1-8B. The random baselines are much less reliable in these two blocks, especially in PPL, so the price-guided rule is not just avoiding a weak fixed default.

The bit-allocation results vary across the two models. On Llama-3.2-3B, the price-guided allocation has the best average downstream score, 53.64, while AMQ has the best PPL. On Llama-3.1-8B, AMQ is stronger across this block, and the price-guided allocation ranks second in average score among the three allocation methods. AMQ nevertheless requires substantially more selection time than our price-guided selector because its search builds and evaluates a quantized model for each candidate allocation.

## 5 Conclusion

This paper formulates weight-space PTQ as pre-deployment configuration selection. Before the output-distribution drift of the completed quantized model can be evaluated, each admissible layer choice must be ranked using its induced layer-output error and deployment cost. Starting from full-precision-to-quantized forward KL, the price theorem shows that score cancellation leaves a quadratic quantization price, with the reference model supplying downstream curvature and each candidate supplying its error covariance. This formulation places finite formats, granularities, codebooks, equivalent transformations, and fixed-geometry bit allocation within the same price-guided search, while the trace reduction provides a practical calibration-time score. The main selector is restricted to weight-space PTQ, and its price predictions rely on a local quadratic approximation, an adequate perturbation moment model, and an aligned curvature estimate. Cross-layer coupling and departures from these conditions delimit its operating regime; extensions beyond weight-space PTQ require additional modeling, as discussed in Appx. A.

## Acknowledgments

This work was supported by the Youth S&T Talent Support Programme of Guangdong Provincial Association for Science and Technology (Grant No. SKXRC2025466). It was also sponsored by CCF-Huawei Populus Grove Fund. We thank Dr. Shuyang Cao at Huawei Taylor Lab. for his valuable advice and suggestions on this work.

## References

[1] Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-free 4-bit inference in rotated LLMs. arXiv:2404.00456, 2024.

[2] 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, pages 7432–7439, 2020.

[3] Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. QuIP: 2-bit quantization of large language models with guarantees. arXiv:2307.13304, 2023.

[4] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

[5] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. arXiv:2208.07339, 2022.

[6] Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. SpQR: A sparse-quantized representation for near-lossless LLM weight compression. arXiv:2306.03078, 2023.

[7] Zhen Dong, Zhewei Yao, Yaohui Cai, Daiyaan Arfeen, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. HAWQ-V2: Hessian aware trace-weighted quantization of neural networks. arXiv:1911.03852, 2019.

[8] Zhen Dong, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. HAWQ: Hessian aware quantization of neural networks with mixed-precision. arXiv:1905.03696, 2019.

[9] Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. Extreme compression of large language models via additive quantization. arXiv:2401.06118, 2024.

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

[11] Elias Frantar, Sidak Pal Singh, and Dan Alistarh. Optimal brain compression: A framework for accurate post-training quantization and pruning. arXiv:2208.11580, 2023.

[12] Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 million context length LLM inference with kv cache quantization. arXiv:2401.18079, 2025.

[13] Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and Xiaojuan Qi. SliM-LLM: Salience-driven mixed-precision quantization for large language models. arXiv:2405.14917, 2024.

[14] M.F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communication in Statistics- Simulation and Computation*, 18:1059–1076, 01 1989.

[15] Jinuk Kim, Marwa El Halabi, Wonpyo Park, Clemens J. S. 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.

[16] 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, 2024.

[17] Deokjae Lee and Hyun Oh Song. Q-Palette: Fractional-bit quantizers toward optimal bit allocation for efficient LLM deployment. arXiv:2509.20214, 2025.

[18] Jung Hyun Lee, Jeonghoon Kim, June Yong Yang, Se Jung Kwon, Eunho Yang, Kang Min Yoo, and Dongsoo Lee. LRQ: Optimizing post-training quantization for large language models by learning low-rank weight-scaling matrices. arXiv:2407.11534, 2025.

[19] Sangjun Lee, Seung-taek Woo, Jungyu Jin, Changhun Lee, and Eunhyeok Park. AMQ: Enabling AutoML for mixed-precision weight-only quantization of large language models. In *Proc. EMNLP*, pages 35532–35550, 2025.

[20] 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.

[21] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for LLM compression and acceleration. arXiv:2306.00978, 2024.

[22] Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

[23] Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. QServe: W4a8kv4 quantization and system co-design for efficient LLM serving. arXiv:2405.04532, 2025.

[24] Han Liu, Haotian Gao, Changya Li, Feng Zhang, Xiaotong Zhang, Wei Wang, and Hong Yu. RUQuant: Towards refining uniform quantization for large language models. arXiv:2604.04013, 2026.

[25] Yifei Liu, Jicheng Wen, Yang Wang, Shengyu Ye, Li Lyna Zhang, Ting Cao, Cheng Li, and Mao Yang. VPTQ: Extreme low-bit vector post-training quantization for large language models. arXiv:2409.17066, 2024.

[26] Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. SpinQuant: LLM quantization with learned rotations. arXiv:2405.16406, 2024.

[27] Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for kv cache. arXiv:2402.02750, 2024.

[28] Yuexiao Ma, Huixia Li, Xiawu Zheng, Feng Ling, Xuefeng Xiao, Rui Wang, Shilei Wen, Fei Chao, and Rongrong Ji. AffineQuant: Affine transformation quantization for large language models. arXiv:2403.12544, 2024.

[29] Vladimir Malinovskii, Andrei Panferov, Ivan Ilin, Han Guo, Peter Richtarik, and Dan Alistarh. Pushing the limits of large language model quantization via the linearity theorem. arXiv:2411.17525, 2024.

[30] Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. arXiv:2004.10568, 2020.

[31] Mohammad Taher Pilehvar and Jose Camacho-Collados. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1128. URL https://aclanthology.org/N19-1128/.

[32] Junbin Qiu, Zhaowei Hong, Renzhe Xu, and Yao Shu. Revisiting zeroth-order Hessian approximation: A single-step policy optimization lens. In *Proc. ICML*, 2026.

[33] 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.

[34] Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. OmniQuant: Omnidirectionally calibrated quantization for large language models. arXiv:2308.13137, 2023.

[35] Sayeh Sharify, Utkarsh Saxena, Zifei Xu, Wanzin Yazar, Ilya Soloveychik, and Xin Wang. Post training quantization of large language models with microscaling formats. arXiv:2405.07135, 2024.

[36] Yao Shu and Zilin Zhu. Compander-aligned query geometry for quantized zeroth-order optimization. arXiv:2605.10673, 2026.

[37] Yao Shu, Shaofeng Cai, Zhongxiang Dai, Beng Chin Ooi, and Bryan Kian Hsiang Low. NASI: Label- and data-agnostic neural architecture search at initialization. In *Proc. ICLR*, 2022.

[38] Yao Shu, Zhongxiang Dai, Zhaoxuan Wu, and Bryan Kian Hsiang Low. Unifying and boosting gradient-based training-free neural architecture search. In *Proc. NeurIPS*, volume 35, pages 33001–33015, 2022.

[39] Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, Xin Jiang, Wulong Liu, and Jun Yao. FlatQuant: Flatness matters for LLM quantization. arXiv:2410.09426, 2024.

[40] Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP#: Even better LLM quantization with hadamard incoherence and lattice codebooks. arXiv:2402.04396, 2024.

[41] Albert Tseng, Qingyao Sun, David Hou, and Christopher De Sa. QTIP: Quantization with trellises and incoherence processing. arXiv:2406.11235, 2024.

[42] Albert Tseng, Zhaofeng Sun, and Christopher De Sa. Model-Preserving Adaptive Rounding. arXiv:2505.22988, 2025.

[43] Mart van Baalen, Andrey Kuzmin, Ivan Koryakovskiy, Markus Nagel, Peter Couperus, Cedric Bastoul, Eric Mahurin, Tijmen Blankevoort, and Paul Whatmough. GPTVQ: The blessing of dimensionality for LLM quantization. arXiv:2402.15319, 2025.

[44] Chenxing Wei, Yao Shu, Ying Tiffany He, and Fei Yu. Flexora: Flexible low-rank adaptation for large language models. In *Proc. ACL*, pages 14643–14682, 2025.

[45] Xiuying Wei, Yunchen Zhang, Xiangguo Zhang, Ruihao Gong, Shanghang Zhang, Qi Zhang, Fengwei Yu, and Xianglong Liu. Outlier suppression: Pushing the limit of low-bit transformer language models. arXiv:2209.13325, 2022.

[46] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. arXiv:2211.10438, 2023.

[47] Bruce Changlong Xu. Activation Sensitivity as a unifying principle for post-training quantization. arXiv:2601.11663, 2026.

[48] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. arXiv:2505.09388, 2025.

[49] Zhewei Yao, Zhen Dong, Zhangcheng Zheng, Amir Gholami, Jiali Yu, Eric Tan, Leyuan Wang, Qijing Huang, Yida Wang, Michael W. Mahoney, and Kurt Keutzer. HAWQ-V3: Dyadic neural network quantization. arXiv:2011.10680, 2021.

[50] Cheng Zhang, Jeffrey T. H. Wong, Can Xiao, George A. Constantinides, and Yiren Zhao. QERA: an analytical framework for quantization error reconstruction. arXiv:2410.06040, 2024.

[51] Haoyu Zhang, Shihao Zhang, Ian Colbert, and Rayan Saab. Provable post-training quantization: Theoretical analysis of OPTQ and Qronos. arXiv:2508.04853, 2025.

[52] Jinhao Zhang, Yunquan Zhang, Daning Cheng, Jun Sun, and Zicheng Yan. CALM: A CKA-guided adaptive layer-wise modularization framework for LLM quantization. arXiv:2512.16282, 2026.

[53] Manyi Zhang, Ji-Fu Li, Zhongao Sun, Haoli Bai, Hui-Ling Zhen, Zhenhua Dong, and Xianzhi Yu. Benchmarking post-training quantization of large language models under microscaling floating point formats. arXiv:2601.09555, 2026.

[54] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models. arXiv:2205.01068, 2022.

[55] Ying Zhang, Peng Zhang, Mincong Huang, Jingyang Xiang, Yujie Wang, Chao Wang, Yineng Zhang, Lei Yu, Chuan Liu, and Wei Lin. QQQ: Quality quattuor-bit quantization for large language models. arXiv:2406.09904, 2024.

[56] Jiaqi Zhao, Miao Zhang, Ming Wang, Yuzhang Shang, Kaihao Zhang, Weili Guan, Yaowei Wang, and Min Zhang. PTQ1.61: Push the real limit of extremely low-bit post-training quantization methods for large language models. arXiv:2502.13179, 2025.

[57] Weibo Zhao, Yubin Shi, Xinyu Lyu, Wanchen Sui, Shen Li, and Yong Li. ASER: Activation smoothing and error reconstruction for large language model quantization. arXiv:2411.07762, 2024.

[58] Yilong Zhao, Chien-Yu Lin, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind Krishnamurthy, Tianqi Chen, and Baris Kasikci. Atom: Low-bit quantization for efficient and accurate LLM serving. arXiv:2310.19102, 2024.

# A Related Work and Scope

**Second-order and reconstruction-based PTQ.** OBQ, GPTQ, and OBC-style methods use local second-order structure to compensate or order weight quantization decisions [10, 11]. QuIP and QuIP# combine incoherence processing with quantization guarantees and codebook structure [3, 40]. Analytical neighbors such as provable OPTQ, QERA, ASER, RUQuant, and activation-sensitivity analysis study local PTQ ingredients or reconstruction refinements [51, 50, 57, 24, 47]. The present formulation treats reconstruction and diagonal scores as reduced prices, retaining downstream curvature when comparing candidate-induced layer-output errors.

**Downstream-aware and model-preserving quantization.** End-loss and model-preservation methods are the closest neighbors in their use of downstream loss or output-distribution information. GuidedQuant uses end-loss guidance for quantization, while YAQA directly targets output-distribution preservation through end-to-end Hessian approximations and Kronecker sketches [15, 42]. Metric-aware adaptive rounding objectives are therefore closely related to the output-side metric in our price. Our contribution operates at the configuration-selection level over admissible weight-space choices: it uses that metric to compare transformations, granularities, quantizer families, and bits through their induced layer-output error covariances and deployment costs.

**Transformations and quantizer families.** LLM.int8(), Outlier Suppression, SmoothQuant, and AWQ show how calibration statistics and outlier ranges can motivate range-aware transformations [5, 45, 46, 21]. OmniQuant, QuaRot, SpinQuant, AffineQuant, and FlatQuant show how rotations and learned transformations change quantization behavior [34, 1, 26, 28, 39]. SpQR, SqueezeLLM, AQLM, GPTVQ, VPTQ, and QTIP expand the admissible weight perturbation set through sparse exceptions, dense-sparse decompositions, additive codebooks, vector quantization, or trellis-coded quantization [6, 16, 9, 43, 25, 41]. Equivalent transformations change the coordinate and resulting quantization error while leaving the full-precision output metric fixed. Related studies in zeroth-order optimization address curvature estimation [32] and perturbation geometry under quantization [36]. Admissible weight-space configurations from these families are compared through their induced layer-output error covariances and deployment costs.

**Mixed precision and deployment constraints.** HAWQ, HAWQ-V2, and HAWQ-V3 use Hessian-aware signals for mixed precision [8, 7, 49]. HIGGS-style methods relate layerwise reconstruction error to perplexity increase and allocate nonuniform quantization levels [29]. Q-Palette takes a rate-distortion view of fractional-bit quantizers after Gaussianizing assumptions [17]. These works provide fixed-geometry allocation references for the broader configuration decision. SliM-LLM, PTQ1.61, QQQ, LRQ, Atom, KIVI, KVQuant, and QServe enlarge the deployment space through salience, low-bit limits, vector or system-aware constraints, KV-cache compression, and serving co-design [13, 56, 55, 18, 58, 27, 12, 23]. Once candidate covariances and deployment costs are available, hardware-aware deployment remains a discrete constrained selection problem. Related layer-selection problems arise in low-rank adaptation [44]. Our contribution is to compare admissible transformations, granularities, quantizer families, and bits through the same priced layer-output error before deployment. Fixed-geometry bit allocation is a special case obtained by holding the other configuration choices fixed; we do not claim novelty for continuous KKT allocation.

**Scope.** The main selector is restricted to weight-space PTQ. Activation quantization, KV-cache quantization, runtime dynamic clipping, and QAT-style learned quantizers can be expressed in the same price formulation only when their effects are represented by a pre-deployment output-error covariance and a deployment cost. The weight-activation extension, the AM-GM balance law within the whitened isotropic surrogate, and the continuous allocation law are formally presented in Appx. B.10 as supplementary consequences of the formulation. They do not define the main selector's objective. The local quadratic price is intended for regimes where low-bit nonlocality, moment-model mismatch, cross-layer coupling, and curvature-proxy error remain secondary to the predicted price gap. Because the selector optimizes only $\hat{\rho}(\alpha)$, these effects delimit the predictor's operating regimes and are not additional optimization terms.

## B Proofs

### B.1 Proof of Thm. 3.1

*Proof.* For each calibration input, the forward KL and the reference-supervised NLL differ by an entropy term independent of the candidate quantized model. Write

$$\mathcal{L}(\theta') \triangleq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[-\sum_y \pi_0(y \mid \mathbf{x}) \log \pi_{\theta'}(y \mid \mathbf{x})\right] . \tag{20}$$

At the full-precision reference,

$$\nabla_{\theta'}\mathcal{L}(\theta')\big|_{\theta'=\theta} = -\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\sum_y \pi_0(y \mid \mathbf{x})\nabla_\theta \log \pi_0(y \mid \mathbf{x})\right] = -\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\sum_y \nabla_\theta \pi_0(y \mid \mathbf{x})\right] = 0 . \tag{21}$$

Taylor expansion around the reference therefore has no first-order term. For a linear layer represented in the candidate coordinate $\mathbf{W}_l(\alpha_l)\mathbf{x}_l(\alpha_l) = \mathbf{W}_l\mathbf{x}_l$, use the per-sample layer-output drift $\mathcal{J}_{\mathbf{x},l}$ from the main text and let $\mathbf{w}_l = \mathrm{vec}(\mathbf{W}_l(\alpha_l))$, $\mathbf{g}_l = \nabla_{\mathbf{z}}\mathcal{J}_{\mathbf{x},l}(\mathbf{z})\big|_{\mathbf{z}=\mathbf{z}_l}$, and $\mathbf{H}_l(\mathbf{x}) = \nabla^2_{\mathbf{z}}\mathcal{J}_{\mathbf{x},l}(\mathbf{z})\big|_{\mathbf{z}=\mathbf{z}_l}$. The second-order chain rule gives

$$\nabla^2_{\mathbf{w}_l}\mathcal{J}_{\mathbf{x},l}(\mathbf{W}_l(\alpha_l)\mathbf{x}_l(\alpha_l)) = \underbrace{(\mathbf{I}\otimes\mathbf{x}_l(\alpha_l))\mathbf{H}_l(\mathbf{x})(\mathbf{I}\otimes\mathbf{x}_l(\alpha_l))^\top}_{\text{Gauss-Newton term}} + \underbrace{\sum_i [\mathbf{g}_l]_i \nabla^2_{\mathbf{w}_l}[\mathbf{z}_l]_i}_{=0} , \tag{22}$$

because each coordinate of $\mathbf{z}_l$ is linear in $\mathbf{w}_l$. Hence the retained block is $\mathbf{H}_l(\mathbf{x}) \otimes \mathbf{x}_l(\alpha_l)\mathbf{x}_l(\alpha_l)^\top$ up to vectorization convention. For $\boldsymbol{\delta}_l = \mathrm{vec}(\boldsymbol{\Delta}_l(\alpha_l))$, the quadratic contribution is

$$\frac{1}{2}\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\mathrm{Tr}\left(\mathbf{H}_l(\mathbf{x})\boldsymbol{\Delta}_l(\alpha_l)\mathbf{x}_l(\alpha_l)\mathbf{x}_l(\alpha_l)^\top\boldsymbol{\Delta}_l(\alpha_l)^\top\right)\right] . \tag{23}$$

Replacing $\mathbf{H}_l(\mathbf{x})$ by the retained average $\mathbf{H}_l$ yields

$$\frac{1}{2}\mathrm{Tr}\left(\mathbf{H}_l\boldsymbol{\Delta}_l(\alpha_l)\mathbf{C}_l(\alpha_l)\boldsymbol{\Delta}_l(\alpha_l)^\top\right) = \frac{1}{2}\mathrm{Tr}\left(\mathbf{H}_l\boldsymbol{\Sigma}_l(\alpha_l)\right) . \tag{24}$$

Let $\mathbf{H}_{lk}$ be the exact Hessian block of $\mathcal{J}$ with respect to vectorized layer weights at the reference, and let $\tilde{\mathbf{H}}_{ll}$ be the retained block satisfying $\frac{1}{2}\boldsymbol{\delta}_l{}^\top\tilde{\mathbf{H}}_{ll}\boldsymbol{\delta}_l = \rho_l(\alpha_l)$. For the stacked perturbation $\boldsymbol{\delta} = (\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_L)$, Taylor's theorem gives

$$\mathcal{J}(\alpha) = \frac{1}{2}\sum_l \boldsymbol{\delta}_l{}^\top\mathbf{H}_{ll}\boldsymbol{\delta}_l + \frac{1}{2}\sum_{l\neq k}\boldsymbol{\delta}_l{}^\top\mathbf{H}_{lk}\boldsymbol{\delta}_k + R_3(\alpha) , \qquad |R_3(\alpha)| \le \frac{M(\alpha)}{6}\|\boldsymbol{\delta}\|_2^3 . \tag{25}$$

The retained block-diagonal quadratic part is therefore

$$\rho(\alpha) = \frac{1}{2}\sum_l \boldsymbol{\delta}_l{}^\top\tilde{\mathbf{H}}_{ll}\boldsymbol{\delta}_l = \sum_l \frac{1}{2}\mathrm{Tr}\left(\mathbf{H}_l\boldsymbol{\Sigma}_l(\alpha_l)\right) , \tag{26}$$

which is (8). The omitted diagnostic term is

$$r(\alpha) = \mathcal{J}(\alpha) - \rho(\alpha) = \frac{1}{2}\sum_l \boldsymbol{\delta}_l{}^\top(\mathbf{H}_{ll} - \tilde{\mathbf{H}}_{ll})\boldsymbol{\delta}_l + \frac{1}{2}\sum_{l\neq k}\boldsymbol{\delta}_l{}^\top\mathbf{H}_{lk}\boldsymbol{\delta}_k + R_3(\alpha) . \tag{27}$$

Equivalently, let $\mathbf{H}_\theta$ be the exact block Hessian with blocks $\mathbf{H}_{lk}$ and let $\tilde{\mathbf{H}}_\theta$ be block diagonal with blocks $\tilde{\mathbf{H}}_{ll}$. Then $\rho(\alpha) = \frac{1}{2}\boldsymbol{\delta}^\top\tilde{\mathbf{H}}_\theta\boldsymbol{\delta}$, so

$$r(\alpha) = \frac{1}{2}\boldsymbol{\delta}^\top(\mathbf{H}_\theta - \tilde{\mathbf{H}}_\theta)\boldsymbol{\delta} + R_3(\alpha) . \tag{28}$$

Cauchy's inequality and the Taylor remainder bound give the compact relative diagnostic used in the main text:

$$|r(\alpha)| \le \frac{1}{2}\|\mathbf{H}_\theta - \tilde{\mathbf{H}}_\theta\|_2\|\boldsymbol{\delta}\|_2^2 + \frac{M(\alpha)}{6}\|\boldsymbol{\delta}\|_2^3 . \tag{29}$$

When $\rho(\alpha) > 0$, dividing this display by $\rho(\alpha)$ gives the relative diagnostic $\eta(\alpha)$ in (9). This proves the main-text price expression and identifies the proof-level terms omitted when the selector uses the quadratic price as its objective. □

### B.2 Finite-Format Moment Bound

*Proof.* Write $\mathbf{\Delta} = \mathbf{\Delta}_l(\alpha_l)$ and $\mathbf{C} = \mathbf{C}_l(\alpha_l)$. Let $\mathbf{h}_l$ and $\mathbf{c}_l$ collect the diagonals of $\hat{\mathbf{H}}_l$ and $\mathbf{C}$, and let $\mathbf{V}_l(\alpha_l)$ be any nonnegative matrix satisfying $\mathbb{E}[\mathbf{\Delta}_{ij}^2] \le [\mathbf{V}_l(\alpha_l)]_{ij}$. Expanding the covariance entrywise gives

$$\mathbb{E}[(\mathbf{\Delta C \Delta}^\top)_{ii'}] = \sum_{j,k} [\mathbf{C}]_{jk} \mathbb{E}[\mathbf{\Delta}_{ij} \mathbf{\Delta}_{i'k}] \,. \tag{30}$$

The independent zero-mean moment model removes all terms except $(i,j) = (i',k)$, hence

$$\mathbb{E}[\hat{\rho}_l(\alpha_l)] = \frac{1}{2} \sum_i [\hat{\mathbf{H}}_l]_{ii} \sum_j [\mathbf{C}_l(\alpha_l)]_{jj} \mathbb{E}[\mathbf{\Delta}_{ij}^2] \le \frac{1}{2} \mathbf{h}_l{}^\top \mathbf{V}_l(\alpha_l) \mathbf{c}_l \,. \tag{31}$$

For the grouped scalar rounding model, $[\mathbf{V}_l(\alpha_l)]_{ij} = (\delta_{g(i,j)}(\alpha_l))^2/12$, which recovers the displayed bit and granularity bound in the main text. □

### B.3 Equivalent Transformation Covariance

*Proof.* Substituting $\tilde{\mathbf{x}} = \mathbf{P}\mathbf{x}$ and $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{P}^{-1}$ gives $\tilde{\mathbf{W}}\tilde{\mathbf{x}} = \mathbf{W}\mathbf{x}$, so the floating-point output and its output-side curvature are unchanged. Quantizing $\tilde{\mathbf{W}}$ by $\tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}$ creates output perturbation $\tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}\tilde{\mathbf{x}}$, whose covariance is $\tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}} \mathbb{E}[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^\top] \tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}{}^\top = \tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}} (\mathbf{P}\mathbf{C}\mathbf{P}^\top) \tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}{}^\top$. Vectorizing the quadratic form gives the Kronecker factor $\mathbf{H} \otimes (\mathbf{P}\mathbf{C}\mathbf{P}^\top)$, making explicit that the transformation moves the input-side metric and perturbation while leaving the output-side price metric fixed. □

### B.4 Proof of the Affine Whitening Optimum

*Proof.* Let $\mathbf{C} = \mathbf{U}\Lambda\mathbf{U}^\top$. If $\mathbf{P} = \sigma\mathbf{Q}\Lambda^{-1/2}\mathbf{U}^\top$ with $\mathbf{Q} \in \mathrm{O}(n)$, then

$$\mathbf{P}\mathbf{C}\mathbf{P}^\top = \sigma^2 \mathbf{Q}\Lambda^{-1/2}\Lambda\Lambda^{-1/2}\mathbf{Q}^\top = \sigma^2 \mathbf{I} \,. \tag{32}$$

Conversely, if $\mathbf{P}\mathbf{C}\mathbf{P}^\top = \sigma^2\mathbf{I}$, then $\mathbf{Q} = \sigma^{-1}\mathbf{P}\mathbf{U}\Lambda^{1/2}$ satisfies $\mathbf{Q}\mathbf{Q}^\top = \mathbf{I}$, so $\mathbf{P} = \sigma\mathbf{Q}\Lambda^{-1/2}\mathbf{U}^\top$. Substituting the whitened covariance into the layer price gives

$$\rho_{\mathbf{P}} = \frac{1}{2} \operatorname{Tr}(\mathbf{H}\tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}(\sigma^2\mathbf{I})\tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}{}^\top) = \frac{\sigma^2}{2} \operatorname{Tr}(\mathbf{H}\tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}\tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}{}^\top) \,. \tag{33}$$

If $|\det \mathbf{P}| = 1$, then $\det(\mathbf{P}\mathbf{C}\mathbf{P}^\top) = \det \mathbf{C}$. For any positive definite matrix $\mathbf{M}$, $\lambda_{\max}(\mathbf{M}) \ge (\det \mathbf{M})^{1/n}$ and $\kappa(\mathbf{M}) \ge 1$. Applying this to $\mathbf{M} = \mathbf{P}\mathbf{C}\mathbf{P}^\top$ gives the lower bounds. Determinant-preserving whitening sets $\mathbf{M} = (\det \mathbf{C})^{1/n}\mathbf{I}$, so it attains both lower bounds. Equality in the $\lambda_{\max}$ lower bound holds only when all eigenvalues of $\mathbf{M}$ are equal, hence exactly when $\mathbf{P}\mathbf{C}\mathbf{P}^\top = (\det \mathbf{C})^{1/n}\mathbf{I}$. The preceding converse gives the displayed form of every optimizer. Taking expectation under $\mathbb{E}[\tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}\tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}{}^\top] = \alpha\mathbf{I} + \mathbf{A}$ yields the stated decomposition with $\varepsilon_{\text{ani}} = (\sigma^2/2)\operatorname{Tr}(\mathbf{H}\mathbf{A})$. Holder's trace inequality gives $|\operatorname{Tr}(\mathbf{H}\mathbf{A})| \le \|\mathbf{H}\|_2 \|\mathbf{A}\|_*$. □

### B.5 Whitening Normalizations and Scale Coupling

The whitening family in Appx. B.4 leaves a scalar degree of freedom. Three standard normalizations fix it. Let $\mathbf{C} = \mathbf{U}\Lambda\mathbf{U}^\top$ with eigenvalues $\lambda_1, \dots, \lambda_n > 0$, and set $\mathbf{P}(\sigma, \mathbf{Q}) = \sigma\mathbf{Q}\Lambda^{-1/2}\mathbf{U}^\top$ with $\mathbf{Q} \in \mathrm{O}(n)$. Then $\mathbf{P}\mathbf{C}\mathbf{P}^\top = \sigma^2\mathbf{I}$, and

$$\sigma_{\text{vol}}^2 = \left( \prod_{i=1}^n \lambda_i \right)^{1/n} , \qquad \sigma_{\text{tr}}^2 = \frac{1}{n} \sum_{i=1}^n \lambda_i \,, \qquad \sigma_{\text{F}}^2 = \frac{n}{\sum_{i=1}^n \lambda_i^{-1}} \,. \tag{34}$$

These are the geometric, arithmetic, and harmonic means induced by the constraints $|\det \mathbf{P}| = 1$, $\operatorname{Tr}(\mathbf{P}\mathbf{C}\mathbf{P}^\top) = \operatorname{Tr}(\mathbf{C})$, and $\|\mathbf{P}\|_F^2 = n$, respectively. Under any of the three normalizations, the whitened weight-only price reduces to

$$\rho_{\mathbf{P}} = \frac{\sigma^2}{2} \operatorname{Tr}(\mathbf{H}\tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}\tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}{}^\top) \,. \tag{35}$$

If the transformed-weight perturbation is isotropic in output space, $\mathbb{E}[\tilde{\boldsymbol{\Delta}}_{\tilde{\mathbf{W}}}\tilde{\boldsymbol{\Delta}}_{\tilde{\mathbf{W}}}{}^{\top}] = \alpha\mathbf{I} + \mathbf{A}$, this becomes

$$\mathbb{E}[\rho_{\mathbf{P}}] = \frac{\sigma^2\alpha}{2}\operatorname{Tr}(\mathbf{H}) + \frac{\sigma^2}{2}\operatorname{Tr}(\mathbf{HA})\,. \tag{36}$$

The scalar $\sigma^2$ is therefore part of the candidate-side coefficient in the implemented statistic, not a replacement for output curvature.

A useful caveat is that, for adaptive scalar quantizers, the transformed-weight distortion can scale with the whitening normalization. For a per-tensor uniform quantizer with range $R_q$ and $b$ bits,

$$\mathbb{E}\left[\|\tilde{\boldsymbol{\Delta}}_{\tilde{\mathbf{W}}}\|_F^2\right] \approx mn\,\frac{R_q^2}{3\cdot 4^{b-1}}\,. \tag{37}$$

If the range follows a homogeneous statistic of $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{P}^{-1} = \sigma^{-1}\mathbf{W}\mathbf{U}\Lambda^{1/2}\mathbf{Q}^{\top}$, then $R_q$ can scale as $1/\sigma$, causing the explicit $\sigma^2$ in the input metric and the implicit $1/\sigma^2$ in distortion to partially cancel. This is a quantizer-model-dependent diagnostic rather than a universal simplification. The implemented selector computes $\hat{\omega}_l$ from the realized perturbation product or from the chosen finite-format moment model.

### B.6 Trace-Score Reduction

*Proof.* Under the scalar input-metric condition $\mathbf{C}_l(\alpha_l) = \sigma_l^2(\alpha_l)\mathbf{I}$, the layer-output error covariance induced by the candidate moment is

$$\boldsymbol{\Sigma}_l(\alpha_l) = \mathbb{E}\left[\boldsymbol{\Delta}_l(\alpha_l)\mathbf{C}_l(\alpha_l)\boldsymbol{\Delta}_l(\alpha_l)^{\top}\right] = \sigma_l^2(\alpha_l)\mathbf{R}_l(\alpha_l)\,. \tag{38}$$

Substituting $\mathbf{R}_l(\alpha_l) = \omega_l(\alpha_l)\mathbf{I} + \mathbf{A}_l(\alpha_l)$ into the layer price gives

$$\rho_l(\alpha_l) = \frac{1}{2}\sigma_l^2(\alpha_l)\operatorname{Tr}\left(\mathbf{H}_l(\omega_l(\alpha_l)\mathbf{I} + \mathbf{A}_l(\alpha_l))\right)\,, \tag{39}$$

which is (16) by linearity of the trace. The second term is the residual left by non-isotropic candidate noise. When a bound is needed, trace duality gives

$$\left|\frac{1}{2}\sigma_l^2(\alpha_l)\operatorname{Tr}\left(\mathbf{H}_l\mathbf{A}_l(\alpha_l)\right)\right| \le \frac{1}{2}\sigma_l^2(\alpha_l)\|\mathbf{H}_l\|_2\|\mathbf{A}_l(\alpha_l)\|_*\,. \tag{40}$$

□

### B.7 Diagonal Scaling Shows What Scaling Cannot Change

For $\mathbf{S} = \operatorname{diag}(s_1,\ldots,s_n)$, define the correlation matrix of $\mathbf{C}$ by

$$[\mathbf{R}_{\mathbf{x}}]_{jk} = \frac{[\mathbf{C}]_{jk}}{\sqrt{[\mathbf{C}]_{jj}[\mathbf{C}]_{kk}}}\,. \tag{41}$$

The transformed covariance is $\mathbf{SCS}$, and its correlation matrix has entries

$$\frac{s_j s_k[\mathbf{C}]_{jk}}{\sqrt{s_j^2[\mathbf{C}]_{jj}\,s_k^2[\mathbf{C}]_{kk}}} = [\mathbf{R}_{\mathbf{x}}]_{jk}\,. \tag{42}$$

Thus diagonal scaling can change variances but not standardized correlations. This is why it cannot remove off-diagonal dependence in $\mathbf{C}$.

### B.8 Rotation Coherence Bounds Scalar-Quantizer Distortion

The main text uses rotations only as a distortion-changing configuration. A simple bound makes that role explicit. For a symmetric per-tensor uniform quantizer with $\mathsf{L}_b = 2^b - 1$, define the coherence of a nonzero matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ by

$$\mu(\mathbf{A}) = \frac{mn\|\mathbf{A}\|_\infty^2}{\|\mathbf{A}\|_F^2}\,. \tag{43}$$

If $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}^\top$ and scalar rounding produces an error $\tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}$, then the standard range bound gives

$$\|\tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}\|_F^2 \le \frac{\mu(\mathbf{W}\mathbf{R}^\top)\|\mathbf{W}\|_F^2}{\mathsf{L}_b^2} \,. \tag{44}$$

Consequently,

$$\rho_{\mathbf{R}}(\tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}) \le \frac{1}{2}\|\mathbf{H}\|_2 \lambda_{\max}(\mathbf{C}) \frac{\mu(\mathbf{W}\mathbf{R}^\top)\|\mathbf{W}\|_F^2}{\mathsf{L}_b^2} \,. \tag{45}$$

The input covariance spectrum is unchanged by $\mathbf{R}$. The bound can improve only through the scalar-quantizer distortion factor.

### B.9 Curvature and Covariance Structure Diagnostics

The supporting diagnostics record three structural observations that determine when the scalarized trace statistic is appropriate. First, output curvature $\mathbf{H}_l$ can be diagonal-dominant without being close to a scalar identity. Writing

$$\mathbf{H}_l = \operatorname{diag}(\mathbf{H}_l) + \mathbf{E}_l \,, \tag{46}$$

the diagonal dominance of $\mathbf{H}_l$ controls the size of $\mathbf{E}_l$, but the entries of $\operatorname{diag}(\mathbf{H}_l)$ can still vary substantially. Replacing $\mathbf{H}_l$ by $\mathbf{I}$ therefore removes output-direction prices. The trace reduction in Equation (16) instead uses $\operatorname{Tr}(\mathbf{H}_l)$ only after the candidate-side output-noise covariance has been reduced to an isotropic matrix.

Second, raw input covariance $\mathbf{C}_l$ can have substantial off-diagonal mass. The diagonal-input reduction

$$\frac{1}{2}\operatorname{Tr}(\mathbf{H}\mathbf{\Delta}_{\mathbf{W}}\operatorname{diag}(\mathbf{C})\mathbf{\Delta}_{\mathbf{W}}{}^\top) \tag{47}$$

is a proxy, not the full price. The diagnostics measure whether $\mathbf{C} - \operatorname{diag}(\mathbf{C})$ behaves like secondary noise for the tested quadratic form. When it does not, the full covariance or a transformation-generated covariance $\mathbf{P}\mathbf{C}\mathbf{P}^\top$ must be priced.

Third, the imatrix or weighted-quantization proxy used in some deployment stacks is exactly the reduced case obtained by setting $\mathbf{H} = \mathbf{I}$ and $\mathbf{C} = \operatorname{diag}(v_1, \ldots, v_n)$:

$$\rho_{\text{imatrix}}(\mathbf{\Delta}_{\mathbf{W}}) = \frac{1}{2}\sum_{j=1}^{n} v_j \|[\mathbf{\Delta}_{\mathbf{W}}]_{:,j}\|_2^2 \,, \qquad v_j = \mathbb{E}[[\mathbf{x}]_j^2] \,. \tag{48}$$

Thus imatrix-style weights preserve diagonal input importance but omit both output curvature and cross-channel input covariance. This is why the main text treats them as reduced proxies after the full price has been defined.

### B.10 Weight-Activation Extension and Reduced Allocation Laws

The main selector addresses weight-space PTQ. We also apply the same output-price operator to a supplementary analysis of weight-activation quantization, while retaining weight-space PTQ as the focus of the main selector.

Fix a layer $\mathbf{z} = \mathbf{W}\mathbf{x}$. Let the weight perturbation be $\mathbf{\Delta}_{\mathbf{W}}$ and write the activation perturbation as $\boldsymbol{\epsilon}_{\mathbf{x}} = \hat{\mathbf{x}} - \mathbf{x}$. Ignoring the product $\mathbf{\Delta}_{\mathbf{W}}\boldsymbol{\epsilon}_{\mathbf{x}}$, the output perturbation is

$$\boldsymbol{\epsilon}_{\mathbf{z}} \approx \mathbf{\Delta}_{\mathbf{W}}\mathbf{x} + \mathbf{W}\boldsymbol{\epsilon}_{\mathbf{x}} \,. \tag{49}$$

The weight-side first-order term is structurally zero at the reference by the same score-cancellation identity used in Thm. 3.1. The activation-side first-order term is different because it vanishes only under a noise model such as $\mathbb{E}[\boldsymbol{\epsilon}_{\mathbf{x}} \mid \mathbf{x}] \approx 0$. Under this conditional zero-mean assumption, and when $\mathbb{E}[\mathbf{x}\boldsymbol{\epsilon}_{\mathbf{x}}{}^\top] \approx 0$, the retained output-error covariance becomes

$$\mathbf{N}_{\text{WA}} = \mathbf{\Delta}_{\mathbf{W}}\mathbf{C}\mathbf{\Delta}_{\mathbf{W}}{}^\top + \mathbf{W}\mathbf{C}_{\boldsymbol{\epsilon}}\mathbf{W}^\top \,, \qquad \mathbf{C}_{\boldsymbol{\epsilon}} \triangleq \mathbb{E}[\boldsymbol{\epsilon}_{\mathbf{x}}\boldsymbol{\epsilon}_{\mathbf{x}}{}^\top] \,, \tag{50}$$

and the corresponding joint price is

$$\rho_{\text{WA}} = \frac{1}{2}\operatorname{Tr}(\mathbf{H}\mathbf{N}_{\text{WA}}) \,. \tag{51}$$

If the conditional zero-mean assumption fails, the omitted first-order bias and the cross moment are not hidden in the selector. They are validity diagnostics for activation quantization in low-bit or strongly clipped regimes.

Under an equivalent input transform $\tilde{\mathbf{x}} = \mathbf{P}\mathbf{x}$, $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{P}^{-1}$, the two retained terms become

$$\rho_W(\mathbf{P}) = \frac{1}{2}\operatorname{Tr}(\mathbf{H}\tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}(\mathbf{P}\mathbf{C}\mathbf{P}^\top)\tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}{}^\top)\,, \qquad \rho_A(\mathbf{P}) = \frac{1}{2}\operatorname{Tr}(\mathbf{H}\mathbf{W}\mathbf{P}^{-1}\mathbf{C}_{\boldsymbol{\epsilon},\tilde{\mathbf{x}}}\mathbf{P}^{-\top}\mathbf{W}^\top)\,. \tag{52}$$

Thus preprocessing has a dual effect in the joint branch. It can make the weight-side input metric more isotropic, while simultaneously changing the activation-noise covariance and the propagation matrix for activation noise. For scale-equivariant per-channel activation quantization, this activation-side effect can cancel in the simplified surrogate. For shared-range or non-equivariant activation quantization, scaling can change activation price.

Inside the whitened isotropic surrogate, let $\mathbf{P}\mathbf{C}\mathbf{P}^\top = \sigma^2\mathbf{I}$, $\mathbb{E}[\tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}\tilde{\mathbf{\Delta}}_{\tilde{\mathbf{W}}}{}^\top] \approx \alpha\mathbf{I}$, and $\mathbf{C}_{\boldsymbol{\epsilon},\tilde{\mathbf{x}}} \approx \beta\mathbf{I}$. Since $\mathbf{P}^{-1}\mathbf{P}^{-\top} = \sigma^{-2}\mathbf{C}$ for the whitening family,

$$\rho_{\mathrm{WA}}(\sigma^2) = \frac{\alpha\sigma^2}{2}\operatorname{Tr}(\mathbf{H}) + \frac{\beta}{2\sigma^2}\Gamma\,, \qquad \Gamma \triangleq \operatorname{Tr}(\mathbf{H}\mathbf{W}\mathbf{C}\mathbf{W}^\top)\,. \tag{53}$$

This scalar branch has the exact AM-GM minimizer

$$\sigma_\star^2 = \left(\frac{\beta\Gamma}{\alpha\operatorname{Tr}(\mathbf{H})}\right)^{1/2}, \qquad \rho_{\mathrm{WA}}(\sigma_\star^2) = \sqrt{\alpha\beta\Gamma\operatorname{Tr}(\mathbf{H})}\,. \tag{54}$$

The equality is exact only after the whitening, isotropic-noise, and conditional-zero-mean reductions have been imposed.

The continuous bit-allocation law is another reduced surrogate. If a layer has retained scalar coefficient $c_l > 0$ and continuous bit variable $b_l$ with reduced price $c_l 2^{-b_l}$, then under the interior constraint $\sum_l b_l = B$,

$$b_l^\star = \bar{b} + \log_2\frac{c_l}{\bar{c}_{\mathrm{geo}}}\,, \qquad \bar{b} = \frac{B}{L}\,, \qquad \bar{c}_{\mathrm{geo}} = \left(\prod_{l=1}^{L} c_l\right)^{1/L}. \tag{55}$$

The optimized retained prices are equal:

$$c_l 2^{-b_l^\star} = \bar{c}_{\mathrm{geo}} 2^{-\bar{b}} \quad \text{for all } l\,. \tag{56}$$

For discrete hardware bitsets, unequal costs, KV-cache constraints, or active lower/upper bounds, this law is a warm start or diagnostic for the budgeted configuration search, not a replacement for the discrete selector in (19).

# C Additional Experiments

## C.1 Empirical Validation of the Price-KL Approximation

Fig. 1 in Sec. 4.1 shows how the price tracks KL drift, which supports its use as a surrogate for evaluating PTQ configurations before deployment. Here, we further identify which approximations account for the remaining gap, including neglecting cross-layer interactions, omitting higher-order Taylor terms, approximating curvature traces, and assuming isotropic input covariance.

**Cross-layer coupling.** Cross-layer coupling occurs when quantization errors in different layers interact, so their joint effect on the output distribution differs from the sum of their individual effects. We test this on OPT-125M and Qwen3-0.6B using 32 WikiText-2 validation sequences of maximum length 512 and weight-only quantization at INT2, INT3, INT4, INT6, and INT8. We compare the joint KL $J_{l,k}$ with the sum $J_l + J_k$ of the separately measured KL values. The overall log-scale correlations in Fig. 2(a,e) are $0.9982$ and $0.9979$, respectively, supporting additive layer scores as a practical approximation.

**Higher-order residuals.** Higher-order residuals cause KL drift to depart from the quadratic behavior predicted by the second-order expansion. Using the same calibration data, we vary the perturbation scale through $\mathbf{W}(\lambda) = \mathbf{W} + \lambda\mathbf{\Delta}_{\mathbf{W}}$, comparing $J(\lambda)/J(1)$ with $\lambda^2$ over $0 < \lambda \leq 1$. The overall Pearson correlations in Fig. 2 (b,f) are $0.9854$ for OPT and $0.9786$ for Qwen, supporting approximately quadratic growth over the tested local range.

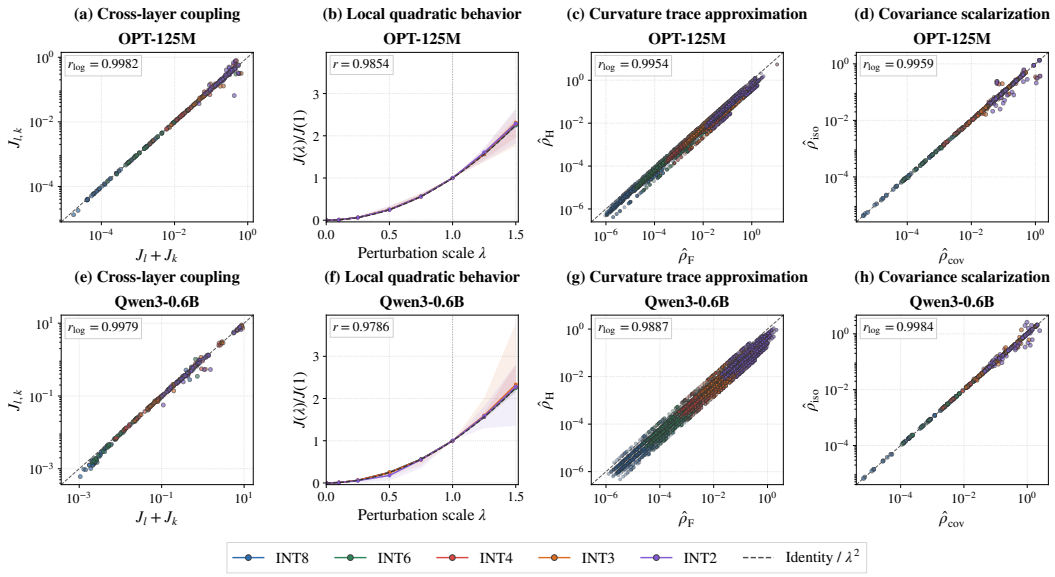


Figure 2: Price-KL approximation diagnostics on OPT-125M (top) and Qwen3-0.6B (bottom). Columns show layer additivity, quadratic KL growth, empirical-Fisher/Hutchinson price comparisons (F/H), and output-normalized covariance-aware/scalar price comparisons (cov/iso). Colors indicate bitwidth; dashed lines denote equality or $\lambda^2$. Scale-sweep lines and bands show medians and interquartile ranges of $J(\lambda)/J(1)$. Annotations give pooled Pearson correlations: $r_{\log}$ on positive log10 pairs, and $r$ between $\lambda^2$ and normalized KL over $0 < \lambda \leq 1$.

**Curvature trace approximation.** The curvature trace weights quantization error by downstream sensitivity, so its estimation affects the resulting price. We compare prices using empirical-Fisher (F) and eight-probe Hutchinson (H) traces of the hard-label language-model loss, keeping candidate output errors fixed and restricting the comparison to scores that are positive for both estimators. The overall log-scale correlations in Fig. 2 (c,g) are $0.9954$ for OPT and $0.9887$ for Qwen, showing similar price trends despite estimator-dependent scaling.

**Anisotropic covariance.** Anisotropic input covariance assigns different importance to different input directions, which is lost when the covariance is replaced by a scalar multiple of the identity. We compare the output-normalized covariance-aware score $\hat{\rho}_{\mathrm{cov}}$, which retains $\mathbf{C}$, with its isotropic counterpart $\hat{\rho}_{\mathrm{iso}}$, keeping the curvature trace fixed. The overall log-scale correlations in Fig. 2 (d,h) are $0.9959$ for OPT and $0.9984$ for Qwen, supporting scalarization as a useful approximation.

# D Experimental Setup

## D.1 Baselines

Here we describe the baselines used in the price-guided configuration selection experiments in Sec. 4.2.

**Bit Allocation.**

- **HIGGS** [29]: HIGGS is a mixed-precision method based on Linearity Theorem. It links per-layer $\ell_2$ reconstruction error to perplexity increase through layerwise linear coefficients, and uses the resulting linearized objective to choose nonuniform layer bitwidths under the same compression budget.
- **AMQ** [19]: AMQ is a search-based mixed-precision weight-only quantization baseline. It formulates layerwise bit allocation as a discrete multi-objective search over bitwidth assignments under a memory budget. An iterative NSGA-II method makes this search practical, after which the selected

allocation is deployed with the final quantization backend. Search-based methods such as AMQ can require substantially more time than our price-guided selector.

**Transformation Selection.**

- **CALM-CKA** [52]: CALM-CKA is a representation similarity baseline adapted from CALM, which evaluates candidate PTQ modules layer by layer and selects the option whose quantized representation best preserves the full-precision representation under linear CKA. In our transformation-selection experiment, we keep the quantizer, bitwidth, and backend fixed and use the same CKA score to choose among admissible pre-quantization transformations.
- **Random Choice**: We randomly sample the admissible transformation candidate from the same candidate set used by the price-guided selector and then quantize.

**Granularity Selection.**

- **Fixed Group-128**: We use a fixed group size of $128$ for all quantized linear layers, keeping the quantizer and bitwidth fixed so that the comparison isolates the value of layerwise granularity selection.
- **Random Choice**: Same as in transformation selection, we randomly sample the admissible granularity candidate from the same candidate set used by the price-guided selector and then quantize.

### D.2 Hyperparameters and Implementation Details

**Common setup.** The KL-alignment study in Fig. 1 uses OPT-125M and Qwen3-0.6B. The configuration-selection experiments in Sec. 4.2 use Llama-3.2-1B, Llama-3.2-3B, and Llama-3.1-8B. For the Llama models, we quantize all transformer linear projections except embeddings, normalization layers, rotary embeddings, and the language-model head, giving $112$, $196$, and $224$ quantized linear layers for the 1B, 3B, and 8B models. Calibration statistics are collected on the WikiText2 validation split with $32$ samples, maximum sequence length $512$, and seed $0$. The KL-alignment run uses the same dataset family with $128$ validation samples of length $1024$.

**Price estimation and bit allocation.** For Llama selection, the downstream curvature term is the Gauss-Newton Fisher trace estimate $\hat{\tau}_l \triangleq \mathrm{Tr}(\hat{\mathbf{H}}_l) = \mathbb{E}\|\nabla_{z_l}\ell\|_2^2$, computed by one forward and backward pass per layer; no Hessian-vector probes are used in the selector. Transformation and granularity candidates are scored by $\hat{\rho}_l(\alpha_l) = \frac{1}{2}\hat{\tau}_l \,\mathrm{MSE}(z_l, \hat{z}_l(\alpha_l))$ on at most $512$ supervised calibration tokens per layer. For bit allocation, each layer chooses from $\{2, 3, 4\}$ bits under the AMQ-compatible budget $3.0 \pm 0.005$, with group size $128$. The price-guided assignment is solved as a binary MILP; HIGGS and AMQ use the same bit set, group size, target budget, and GPTQ backend. GPTQ quantization uses WikiText2 calibration data, true-sequential quantization, activation ordering, and no MSE search.

**Candidate sets.** Transformation selection fixes W4 quantization with group size $128$ and chooses among the identity transform, AWQ-style scaling, SmoothQuant-style scaling, Hadamard rotation, and normalized diagonal whitening. The SmoothQuant-style scale uses $\alpha = 0.5$. When the input dimension is a power of two, Hadamard rotations use deterministic layerwise Rademacher signs with seed $0$. The CALM-CKA baseline chooses the candidate with maximum linear CKA, and random baselines use seeds $0$, $1$, and $2$. Granularity selection fixes W4, no transformation, uniform integer quantization, and min-max clipping, and chooses among per-tensor, per-channel, group sizes $32$, $64$, and $128$, and $64 \times 64$ block-wise quantization. Granularity costs include FP32 scale metadata, reported as effective weight bits; ties within $1\%$ of the best price are broken by lower effective bit cost and then by the latency proxy. Transformation and granularity checkpoints are saved as dense fake-quant Hugging Face checkpoints rather than packed deployment artifacts.

**Evaluation protocol.** WikiText2 PPL is evaluated on the test split using full-text concatenation with slow tokenization, sequence length $2048$, $141$ sequences, and $288{,}627$ supervised tokens. Downstream accuracy is evaluated with LightEval, batch size $32$, greedy decoding with temperature $0$ and seed $42$. The five tasks are BoolQ [4], TruthfulQA MC2 [22], PIQA [2], WinoGrande [33],

and WiC [31]; the reported average is the arithmetic mean of the five percentages. FP16 references use float16.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the paper as pre-deployment configuration selection using priced layer-output error and distinguish the validation protocol from completed end-to-end deployment claims.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appx. A discusses the weight-space scope, local quadratic approximation, moment-model mismatch, cross-layer coupling, curvature-proxy mismatch, and operating regimes.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Refer to Sec. 3 for the full set of assumptions and Appx. B for the complete proof.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All the information is provided in Sec. 4.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All the data and code are provided in the supplemental material.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All information is provided in Sec. 4.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All evaluation results are reported with the average of multiple independent runs.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: All information is provided in Sec. 4.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We provided the discussion in Appx. A.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a new model or dataset with high misuse risk.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All code and data are open-sourced.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce or release a new dataset, model, or software asset in the current submission.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing experiments or research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The research does not involve research with human subjects.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: We utilize LLM only for writing, editing, and formatting purposes.