BEYOND RECONSTRUCTION: FISHER-REGULARIZED POST-TRAINING QUANTIZATION FOR LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models must be compressed to be deployed under realistic memory and latency budgets, and post-training quantization (PTQ) is dominant because it needs no retraining. Existing PTQ methods minimize a per-layer reconstruction error that approximates perplexity, but this proxy is not aligned with downstream task loss. Since a deployed model is typically dedicated to one task, the most direct way to preserve its task accuracy is to draw the calibration data from that task, yet this does not close the gap. We argue that the mismatch lies in the objective rather than in the data, and we introduce a simple yet effective modification that augments the PTQ objective with a Fisher-information regularizer preserving the weight directions to which the loss is most sensitive. We instantiate the regularizer in two forms, a *Task Fisher* computed from a labeled task training split and a *Generic Fisher* computed from the generic calibration data already used by PTQ, the latter removing any dependence on task labels and yielding a single quantized model that serves all tasks rather than a separate quantization per task. The regularizer requires no architectural changes and integrates into any per-layer PTQ solver as a diagonal loading of the per-layer curvature. Across five LLMs from 1B to 70B parameters and 2–4 bit widths, Task Fisher and Generic Fisher improve zero-shot accuracy and perplexity to a comparable degree, showing that neither task labels nor per-task quantization is necessary. Adding the label-free Generic Fisher to standard PTQ solvers raises multiple-choice accuracy by up to 6.3 points and generative-task accuracy by up to 9.4 points over the unregularized baselines, and brings the same improvement to a state-of-the-art solver.

1 INTRODUCTION

Large language models deliver strong performance across many tasks (Achiam et al., 2023), but their memory footprint and inference cost make deployment expensive (Chen et al., 2024). Weight quantization is the most practical way to reduce both, and among the available techniques post-training quantization (PTQ) is especially attractive because it requires no retraining and composes with existing pipelines (Frantar et al., 2023; Lin et al., 2024; Chee et al., 2023; Yao et al., 2022). Its dominant form is layer-wise PTQ, which quantizes a model one layer at a time using a small set of generic calibration data to minimize the output error that quantization introduces in each layer. This recipe has become the standard way to compress LLMs to 2–4 bits without the training data.

In practice, a quantized model is often deployed for a specific task, and what matters is that quantization preserves accuracy on that task rather than on text in general. Yet the standard recipe does not optimize for this. The reconstruction error that layer-wise PTQ minimizes is only a local proxy for perplexity (PPL) on generic text (Malinovskii et al., 2025), and more fundamentally it discards information about how weight perturbations move the loss the model is meant to minimize. The PTQ objective therefore attends only to reconstruction while omitting the information that governs task accuracy. This raises the question of how PTQ can limit accuracy degradation on a target task.

The diagnosis above attributes the problem to the objective, but a simpler and more direct hypothesis places it in the data. Because the standard recipe uses generic calibration data whose distribution does not reflect the task, the most direct fix is to draw the calibration data from the target task itself, so that the reconstruction error is measured on task-relevant inputs and quantization better preserves behavior on the deployment distribution. This is the minimal intervention on the visible mismatch,

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

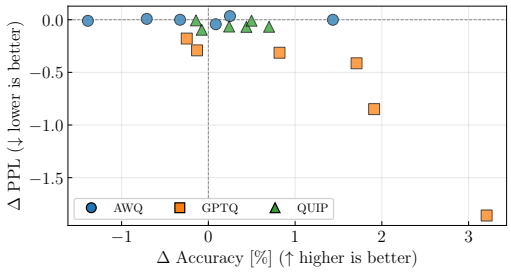

Figure 1: Perplexity and accuracy do not move together under task-matched calibration. Each point is one of six zero-shot tasks quantized with one of three PTQ solvers (AWQ, GPTQ, QuIP) at 4 bits on Llama-3-8B, and reports the change relative to the generic-calibration baseline when the calibration data is switched to the target task. The perplexity change (y-axis, lower is better) and the accuracy change (x-axis, higher is better) are uncorrelated.

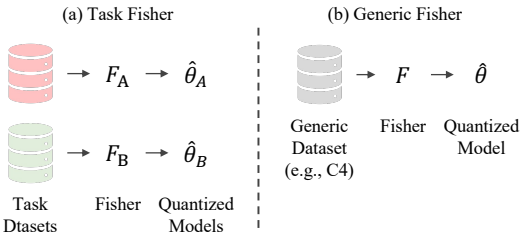

Figure 2: Overview of the two Fisher estimators. (a) Task Fisher computes a Fisher matrix F_t from each task's labeled training split under the task loss, and produces one quantized model $\hat{\theta}_t$ per task. (b) Generic Fisher computes a single Fisher matrix from the generic calibration data under a cross-entropy loss, and produces one quantized model that serves all tasks. Both add the same Fisher regularizer to the per-layer objective, leaving the PTQ solver unchanged.

and it leaves the objective untouched. However, Williams & Aletras (2024) benchmark this idea on more than forty datasets and report that task-data calibration does not consistently improve task accuracy. Their study reports the outcome but does not isolate its cause, leaving open whether task data is uninformative or whether the objective cannot exploit it.

To separate these, we revisit task-matched calibration on the Llama-3-8B (Grattafiori et al., 2024) with three representative layer-wise solvers, and alongside task accuracy we measure perplexity on the task text itself, the quantity the objective targets. As shown in fig. 1, task-matched calibration tends to improve, or at least not harm, perplexity, whereas its effect on accuracy scatters in both directions. The objective improves on its own target when given task data, so the data is informative, yet this gain does not carry over to task accuracy. The bottleneck therefore lies in the objective rather than in the data, and improving task accuracy requires a modification at the level of the objective.

Directly minimizing the task loss inside a PTQ solver is infeasible, because quantizing billions of weights to a few discrete levels is combinatorial, and because layer-wise PTQ deliberately avoids the end-to-end gradient-based training that optimizing a global task loss would require. Task information must therefore be introduced through some relaxation. We propose a simple yet effective modification that augments the reconstruction error with a Fisher-weighted regularization term (Kirkpatrick et al., 2017), where the Fisher information yields the second-order Taylor approximation of the loss increase under quantization and is the relaxation that produces a quadratic weight-space penalty compatible with any layer-wise PTQ solver. This term restores the loss-curvature information that pure reconstruction discards, and we instantiate it in two forms. *Task Fisher* computes the regularizer from a labeled task training split under the task loss, which is the most direct realization but requires such a split for each target task and must be recomputed for each new task. *Generic Fisher* computes the regularizer from the generic calibration data under a cross-entropy loss and requires no labels, removing the dependence on task data that limits Task Fisher. Because it only adds a term to the objective, the regularizer is solver-agnostic, reduces to the standard solver when it is switched off, and drops into existing layer-wise PTQ solvers without architectural changes.

Experiments on Llama-3-8B show that both Task Fisher and Generic Fisher improve task accuracy on standard layer-wise solvers, and that Generic Fisher extends the same gain to a recent state-of-the-art solver. Both estimators also improve perplexity. The improvements hold across models of different sizes and families, from 1B to 70B. That the label-free Generic Fisher matches Task Fisher, purely by modifying the objective, supports the diagnosis that the mismatch lies in the objective rather than the data, and indicates that reconstruction omits broadly important loss directions rather than that task specialization is required. The overhead is small, because the Fisher requires only a single additional forward-backward sweep per model, is computed once and cached, and leaves the underlying PTQ solver unchanged.

Our main contributions are as follows.

- We identify the source of the problem, showing through a preliminary study on a recent model and three layer-wise solvers that task-matched calibration improves perplexity but does not propagate to task accuracy, which locates the mismatch in the objective rather than in the data.
- We propose an objective-level modification that adds a Fisher-weighted regularizer to the reconstruction error, realized as a Task Fisher that uses task data and a label-free Generic Fisher, both of which integrate into any per-layer PTQ solver without architectural changes, as a generalization that recovers the standard solver when the regularizer is switched off, and supported by a Taylor-expansion interpretation.
- We demonstrate that both Task and Generic Fisher improve zero-shot accuracy and perplexity over the baselines across three solvers and 2–4 bit widths, and that the label-free Generic Fisher scales to five LLMs from 1B to 70B and to a state-of-the-art solver at small cost.

2 Related Work

2.1 Layer-Wise Post-Training Quantization for LLMs

Layer-wise PTQ compresses an LLM without retraining by minimizing the output error that quantization introduces over one local region at a time, whether a single layer or a small block of layers, and solvers differ mainly in how they precondition or restrict this reconstruction objective. One family uses second-order information, ordering and correcting the weight updates through the per-layer Hessian or its inverse, often after rotations that reduce the incoherence between the weights and the Hessian (Liu et al., 2025). Rooted in classical second-order saliency (LeCun et al., 1989; Hassibi & Stork, 1992), this line now spans a range of LLM solvers (Frantar et al., 2023; Chee et al., 2023; Tseng et al., 2024). A second family is activation-aware, rescaling or smoothing the channels that activation statistics flag as hard to quantize, or learning such transformations end to end (Yao et al., 2022; Xiao et al., 2023; Lin et al., 2024; Shao et al., 2024; Wang et al., 2026). Despite their differences, all of these solvers minimize a reconstruction loss that is known to track perplexity (Malinovskii et al., 2025) rather than the loss of any specific task, even though they are now routinely evaluated on downstream task accuracy as well (Gong et al., 2024; Zhao et al., 2025). Rather than proposing a new solver, our work augments this shared objective with a term that curbs the task-accuracy degradation the reconstruction loss alone leaves unchecked, so it builds on these methods and composes with all of them.

2.2 Task-Specialized Quantization

To make quantization sensitive to a target task, prior work intervenes either on the calibration data or on the allocation of precision across weights. At the data level, studies revisit which data to calibrate on. The choice of calibration data can substantially affect downstream performance (Williams & Aletras, 2024), though this effect diminishes on modern LLMs (Paglieri et al., 2024), while later work synthesizes or regenerates the calibration data itself (Williams et al., 2025; Xiao et al., 2026). At the weight level, TAQ and TaCQ identify the layers or weights that are important to a task and assign them higher precision (LeVi et al., 2025; Xiao et al., 2025). These approaches leave the per-layer objective unchanged, adapting instead the inputs to that objective or the bit-allocation policy, the latter requiring non-uniform precision. We instead modify the per-layer objective itself at uniform precision, which runs on existing quantized-inference kernels without the extra indexing and mixed-width decoding that non-uniform allocation needs to realize its savings on hardware. Because our intervention acts at a different point, it is complementary to both lines and can be combined with task-specific calibration data or with mixed-precision allocation when supported.

2.3 Quantization Objectives

Several methods target the quantization objective itself, either deriving it from a second-order view of the loss or reshaping it with curvature information. AdaRound casts weight rounding as a task-loss Taylor term that reduces to a layer-wise reconstruction objective (Nagel et al., 2020), loss-aware quantization relates the quantization step size to the curvature of the loss surface (Nahshan et al.,

2021), and BRECQ weights its block-wise reconstruction by the Fisher information of the network output (Li et al., 2021). A related group applies curvature not to the objective but to the quantizer or the bit budget, weighting a non-uniform quantizer by per-weight Fisher information (Kim et al., 2024) or driving group-wise mixed-precision allocation from Hessian-based salience (Huang et al., 2025). Recent work improves the reconstruction objective in a solver-agnostic way by propagating and compensating the quantization error across layers (Arai & Ichikawa, 2025). Our work also acts on the per-layer objective and is solver-agnostic, but rather than keeping a reconstruction metric that tracks perplexity, it adds the Fisher of a chosen loss as an additive term that steers the objective toward that loss and vanishes when disabled, recovering the standard solver exactly (see appendix), unlike Fisher-folding methods such as BRECQ, whose per-output weighting breaks the shared Hessian that closed-form solvers factorize once.

3 PRELIMINARIES

3.1 LAYER-WISE POST-TRAINING QUANTIZATION

We denote the weight matrix of one linear layer by $W = (w_{ij}) \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, its quantized counterpart drawn from a discrete codebook by $\hat{W}$, and the quantization error by $\Delta = \hat{W} - W$. The standard layer-wise PTQ objective minimizes the activation reconstruction error

$$\mathcal{L}_{\text{recon}}(\hat{W}) = \big\|(\hat{W} - W)X\big\|_F^2 = \operatorname{tr}\big(\Delta\, \Sigma_x\, \Delta^\top\big), \tag{1}$$

where $X = (x_{jn}) \in \mathbb{R}^{d_{\text{in}} \times N}$ stacks the N calibration input activations and $\Sigma_x = XX^\top$ acts as the per-layer Hessian, $\|\cdot\|_F$ is the Frobenius norm, and tr is the trace of a matrix.

Concrete solvers differ in how they restrict or precondition this optimization, for instance through greedy updates based on Σ_x^{-1}, a low-dimensional family of per-channel rescalings, or an orthogonal preconditioning of the statistics, and minimize the reconstruction objective in eq. (1). Solvers that jointly quantize a block of layers instead minimize the analogous reconstruction error of the block output, a more general form for which eq. (1) is the single-linear special case.

3.2 FISHER INFORMATION OF THE SURROGATE LOSS

Given a surrogate loss $\mathcal{L}$, we define the per-channel Fisher diagonal of the layer as

$$f_j = \mathbb{E}\left[\left\|\frac{\partial \mathcal{L}}{\partial y}\right\|^2 \cdot x_j^2\right], \qquad F = \operatorname{diag}(f_1, \ldots, f_{d_{\text{in}}}), \tag{2}$$

where $y \in \mathbb{R}^{d_{\text{out}}}$ is the layer output, $x_j \in \mathbb{R}$ is the j-th input channel, and $\operatorname{diag}(\cdot)$ denotes the diagonal matrix with the given entries. The expectation is over the data distribution that defines $\mathcal{L}$. Intuitively, f_j measures how strongly the surrogate loss responds to a change in the input channel j, so it acts as a per-channel importance that specifies which channels a quantizer should perturb least. Storing or inverting the full layer Fisher is quadratic in the layer width, so we keep only its per-channel diagonal, which captures each channel's sensitivity at the cost of a single forward-backward pass per minibatch. This diagonal is the empirical Fisher information, which agrees with the generalized Gauss-Newton (equivalently, the true Fisher) matrix for exponential-family losses with labels sampled from the model, and is standard in the EWC line of work. The choice of the loss $\mathcal{L}$ and the data on which it is evaluated parameterizes a family of estimators.

4 METHOD

The reconstruction objective in eq. (1) tracks the increase of perplexity, but it is not in general aligned with the loss of any specific downstream task (Malinovskii et al., 2025). Substituting task data for calibration data preserves the form of this proxy and therefore does not, on its own, target the task loss, so the misalignment is at the objective level rather than the data level. We therefore modify the objective itself, injecting task-loss curvature into the per-layer optimization through a Fisher-weighted regularizer that acts purely at the objective level and leaves the PTQ solver unchanged.

We develop this idea as follows. We first define the Fisher-regularized per-layer objective, which reduces to standard PTQ when the regularization weight is zero and otherwise adds a Fisher term

governed by two design choices, the loss used to estimate the Fisher information and the data on which it is estimated. These choices yield two estimators shown in fig. 2, Task Fisher, computed from a labeled task training split under the task loss, and Generic Fisher, computed from the generic calibration data under a cross-entropy loss, which we adopt as the default. We then describe how the regularizer integrates into existing layer-wise PTQ solvers, the normalization that makes it scale-invariant and numerically stable, and close with a theoretical interpretation of the regularizer as a second-order surrogate for the task-loss degradation.

4.1 Fisher-Regularized PTQ Objective

Minimizing the task loss directly inside the solver is infeasible, because selecting discrete codebook entries for billions of weights is a combinatorial optimization, and because layer-wise PTQ deliberately avoids the end-to-end gradient-based training that optimizing a global task loss would require, so we introduce task information through a relaxation. We use the second-order Taylor approximation of the task loss, its Fisher information, which is the relaxation that yields a quadratic weight-space penalty compatible with the per-layer reconstruction objective, regardless of how each solver optimizes it. We augment the per-layer objective with a Fisher-weighted quadratic penalty, giving the Fisher-Regularized PTQ (FR-PTQ) objective,

$$\mathcal{L}_{\mathrm{FR}}(\hat{W}) = \underbrace{\mathcal{L}_{\mathrm{recon}}(\hat{W})}_{\text{solver's reconstruction}} + \lambda \underbrace{\sum_j f_j \left\|\hat{W}_{:,j} - W_{:,j}\right\|_2^2}_{\text{Fisher reg.}}, \tag{3}$$

where $\mathcal{L}_{\mathrm{recon}}$ is the reconstruction objective that the chosen layer-wise solver already minimizes, and the second term is our contribution, a Fisher regularizer that scales each weight's squared change by the importance f_j of its input channel. Because it depends only on the weights, not on the solver's reconstruction structure, it composes with any layer-wise solver. The regularizer modifies only the objective, not the solver. At $\lambda = 0$, it reduces to standard PTQ for any solver, so the formulation is a generalization (see appendix). For the canonical single-linear reconstruction $\mathcal{L}_{\mathrm{recon}} = \mathrm{tr}(\Delta\Sigma_x\Delta^\top)$ in eq. (1), the regularizer is the quadratic form $\sum_j f_j\|\Delta_{:,j}\|_2^2 = \mathrm{tr}(\Delta F\Delta^\top)$, so the two terms merge into a single effective Hessian,

$$\mathcal{L}_{\mathrm{FR}}(\hat{W}) = \mathrm{tr}\big(\Delta H_{\mathrm{eff}}\Delta^\top\big), \qquad H_{\mathrm{eff}} = \Sigma_x + \lambda F. \tag{4}$$

Therefore, for solvers that expose Σ_x, the regularizer acts exactly as a diagonal loading of the per-layer Hessian, while solvers that optimize a different reconstruction surrogate add the same Fisher term to their own objective, as we detail below. The design space of the regularizer therefore reduces to two choices, the loss $\mathcal{L}$ used to estimate F and the data on which it is estimated. These two axes define the estimators presented in the following subsections. We preprocess the Fisher diagonal by per-layer mean-normalization, clipping at a threshold c_F, and re-normalization, which makes λ a scale-invariant hyperparameter and bounds the condition number of H_{eff} (see appendix).

4.2 Task Fisher

The most direct estimator computes F from a labeled task training split under the task loss. For a multiple-choice task with context c, K options where option k is the token sequence $o_{k,1:T_k}$ and $T_k \in \mathbb{N}$ is its token length, and correct index $y \in \{1, \ldots, K\}$, we score each option by its log-likelihood s_k and use the K-way classification loss. For a prompt-answer task with prompt p and answer tokens $a_{1:M}$, we use a token-level cross-entropy on the answer tokens with the prompt masked out,

$$s_k = \sum_{t=1}^{T_k} \log p_\theta(o_{k,t} \mid c, o_{k,<t}), \quad \mathcal{L}_{\mathrm{MC}} = -\log \frac{e^{s_y}}{\sum_{k=1}^{K} e^{s_k}}, \tag{5}$$

$$\mathcal{L}_{\mathrm{PA}} = -\sum_{t=1}^{M} \log p_\theta(a_t \mid p, a_{<t}). \tag{6}$$

This estimator is the most faithful to the loss we wish to protect, and we treat it as the methodological starting point of our formulation. It has two practical limitations, however. First, it requires a labeled training split for the target task, which is unavailable when the downstream task is not known in advance or when labels are costly to obtain. Second, it produces a separate Fisher per task, so the regularizer must be recomputed and the model requantized for each new task. These limitations motivate an estimator that retains curvature-aware regularization without depending on task labels.

4.3 GENERIC FISHER

To overcome both limitations of Task Fisher, Generic Fisher computes F from the same generic calibration data that already feeds Σ_x, under a cross-entropy loss. Its premise is that the knowledge an LLM acquires during pretraining is broadly useful, so protecting the weight directions that this generic loss is most sensitive to protects the model's task-relevant knowledge as well. Generic Fisher requires no labels, reuses a single Fisher across all tasks, and shares its data source with the PTQ calibration data, so the activation statistics underlying Σ_x and F are consistent. For a sequence $x_{1:T}$, we evaluate the cross-entropy (CE) at the final token with the preceding tokens as context,

$$\mathcal{L}_{\mathrm{CE}} = -\log p_\theta(x_T \mid x_{<T}), \tag{7}$$

which matches the form of zero-shot evaluations that condition on a context and score a candidate continuation, and is the minimal departure from the loss used by Task Fisher.

4.4 INTEGRATION WITH EXISTING PTQ SOLVERS

Because the regularizer modifies only the per-layer objective, it composes with any layer-wise PTQ solver, independently of whether F comes from Task Fisher or Generic Fisher. Solvers whose update depends on the second-order statistics Σ_x and its inverse absorb the regularizer as a diagonal loading, replacing Σ_x with H_{eff} from eq. (4) without changing the form of the update. Solvers that instead optimize the reconstruction by other means, whether over a restricted parametric family of transformations or by gradient descent over a block or window of layers, and thus do not expose Σ_x, absorb the regularizer as an additional Fisher-weighted term added per constituent linear layer to their objective. This loss-level integration constrains how much the regularizer can reshape the solution, which we revisit in the experiments. The solver-specific details are deferred to the appendix.

4.5 THEORETICAL INTERPRETATION

Under standard regularity assumptions on the loss $\mathcal{L}$ used to define F and approximate stationarity at the pretrained weights, the Fisher penalty $\mathrm{tr}(\Delta F \Delta^\top)$ coincides with the second-order Taylor approximation of the degradation of $\mathcal{L}$ induced by the quantization perturbation Δ. This identifies the Fisher penalty as a principled second-order surrogate for that degradation, so minimizing it directly targets $\mathcal{L}$ rather than the perplexity proxy that the reconstruction term tracks, a local approximation near the pretrained weights that we use as a design principle rather than a global guarantee. We state the formal propositions, including the special cases, the second-order surrogate identity, the QuIP compatibility result, the AWQ-family lower bound, and the conditioning bound, in the appendix.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

We evaluated five LLMs spanning two families and four parameter scales, Llama-3.2-1B, Llama-3.2-3B, Llama-3-8B, Llama-3-70B (Grattafiori et al., 2024), and Qwen3-8B (Yang et al., 2025). Most experiments integrated our regularizer into three representative PTQ solvers, GPTQ (Frantar et al., 2023), AWQ (Lin et al., 2024), and QuIP (Chee et al., 2023), while a subset additionally applied it to SliderQuant (Wang et al., 2026) on Llama-3-8B. We quantized weights uniformly at $W \in \{2, 3, 4\}$ bits with full-precision (FP16) activations. The calibration data for all solvers consisted of 128 randomly sampled segments of 2048 tokens each drawn from the C4 dataset (Raffel et al., 2020). We estimated Fisher information from a separate draw of 1024 segments of 2048 tokens each. Generic Fisher reuses the C4 dataset with the last-token CE loss by default, whereas Task Fisher draws from the training split of each evaluation task, using the multiple-choice log-likelihood loss for the multiple-choice tasks and a prompt-masked CE loss for the non-multiple-choice tasks. We fixed the clipping threshold to $c_F = 10$ and swept $\lambda \in \{0.1, 1.0, 2.0\}$. We selected the λ that minimized C4 perplexity, and report it together with the unregularized $\lambda = 0$ baseline. We report perplexity averaged over WikiText-2 (Merity et al., 2017) and PTB (Marcus et al., 1994), holding out C4 from evaluation because it serves as the calibration data and the criterion for selecting λ, and mean multiple-choice (MC) zero-shot accuracy over ARC-easy, ARC-challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), RACE (Lai et al., 2017), and

Table 1: Main results on Llama-3-8B under C4 calibration. A dagger (†) marks a task-independent PPL, computed with the HellaSwag task Fisher. Within each (bit, method) block, bold marks the better of the baseline and a proposed estimator in each column (higher for MC and Non-MC, lower for PPL). The two proposed estimators are not ranked against each other. A dash (–) in the PPL column marks a perplexity of 1000 or above, which we treat as diverged and omit as not meaningful. SliderQuant is reported with Generic Fisher only, since per-task requantization is prohibitive for this costly solver. The full per-task breakdown is in the appendix.

	INT2			INT3			INT4		
Method	MC (↑)	Non-MC (↑)	PPL (↓)	MC (↑)	Non-MC (↑)	PPL (↓)	MC (↑)	Non-MC (↑)	PPL (↓)
FP16	63.85	49.26	8.36	63.85	49.26	8.36	63.85	49.26	8.36
AWQ	32.52 ± 0.00	**0.00 ± 0.00**	–	**59.91 ± 0.00**	36.68 ± 0.00	11.24 ± 0.00	63.27 ± 0.00	46.33 ± 0.00	8.85 ± 0.00
w/Task	**32.62 ± 0.05**	**0.00 ± 0.00**	–	59.68 ± 0.02	**38.30 ± 0.02**	**10.75 ± 0.00**†	**63.31 ± 0.03**	**46.71 ± 0.00**	**8.81 ± 0.00**†
w/Generic	32.46 ± 0.03	**0.00 ± 0.00**	–	59.65 ± 0.06	**37.71 ± 0.08**	**10.76 ± 0.00**	**63.29 ± 0.03**	**46.46 ± 0.04**	**8.81 ± 0.00**
GPTQ	32.94 ± 0.29	0.00 ± 0.00	–	33.12 ± 0.08	0.16 ± 0.06	–	58.23 ± 0.86	30.30 ± 1.76	12.05 ± 0.37
w/Task	**32.99 ± 0.11**	**0.01 ± 0.00**	–	**37.88 ± 0.13**	**2.32 ± 0.17**	**104.94 ± 11.24**†	**61.48 ± 0.14**	**40.31 ± 0.37**	**10.23 ± 0.04**†
w/Generic	32.90 ± 0.19	**0.00 ± 0.00**	–	**38.14 ± 0.17**	**2.02 ± 0.23**	**109.34 ± 12.70**	**61.57 ± 0.07**	**39.69 ± 0.34**	**10.23 ± 0.04**
QuIP	34.90 ± 0.42	1.30 ± 0.23	169.03 ± 30.90	59.06 ± 0.33	30.84 ± 1.19	10.79 ± 0.18	62.27 ± 0.32	43.19 ± 1.02	9.33 ± 0.11
w/Task	**35.33 ± 0.18**	**1.65 ± 0.12**	**103.29 ± 4.41**†	**60.00 ± 0.27**	**34.27 ± 0.71**	**10.39 ± 0.09**†	**62.55 ± 0.14**	42.16 ± 0.72	**9.15 ± 0.03**†
w/Generic	**35.52 ± 0.17**	**1.83 ± 0.20**	**112.58 ± 8.14**	**59.65 ± 0.30**	**32.33 ± 1.14**	**10.39 ± 0.05**	**62.75 ± 0.14**	**43.25 ± 0.63**	**9.16 ± 0.04**
SliderQuant	37.66 ± 0.15	0.68 ± 0.05	880.29 ± 66.05	57.00 ± 0.15	16.32 ± 0.41	15.12 ± 0.19	62.64 ± 0.10	21.43 ± 0.12	9.40 ± 0.02
w/Generic	**39.43 ± 0.41**	**1.03 ± 0.08**	**325.17 ± 9.26**	**57.82 ± 0.35**	**16.55 ± 0.38**	**13.24 ± 0.11**	**62.65 ± 0.13**	**22.94 ± 1.14**	**9.27 ± 0.02**

WinoGrande (Sakaguchi et al., 2021) through the LM evaluation harness (Gao et al., 2024). For the in-depth analysis on Llama-3-8B we additionally evaluated three non-multiple-choice (Non-MC) tasks, a last-word prediction task (LAMBADA (Paperno et al., 2016)) and two free-form QA tasks (GSM8K (Cobbe et al., 2021) and TriviaQA (Joshi et al., 2017)). We scored each task with its standard metric, using last-word prediction accuracy for LAMBADA and exact match for GSM8K and TriviaQA, and report the exact metric definitions in the appendix. Each reported value is the mean $\pm$ standard error of the mean over five random seeds.

5.2 Main Results on Llama-3-8B

We examined on Llama-3-8B whether adding the Fisher regularizer improves zero-shot task accuracy and perplexity over the unregularized baseline, and whether Generic Fisher matches Task Fisher. Table 1 reports MC and Non-MC zero-shot accuracy and perplexity for the two estimators and the λ=0 baseline across GPTQ, AWQ, and QuIP at 2–4 bits, and for Generic Fisher applied to SliderQuant, all under C4 calibration.

For GPTQ and QuIP the regularizer improves MC and Non-MC accuracy over the baseline in most settings, with the largest gains for GPTQ, whose baseline is the most degraded, and it also lowers perplexity wherever perplexity remains meaningful. AWQ improves by the smallest margin, with gains mainly in Non-MC accuracy and perplexity while its MC accuracy stays roughly flat. On SliderQuant, Generic Fisher improves MC and Non-MC accuracy and lowers perplexity at all three bit widths. Task Fisher and Generic Fisher are close across cells, with no consistent winner between them. At 2 bits the AWQ and GPTQ baselines collapse and the regularizer does not recover them, whereas for QuIP and SliderQuant it improves accuracy and more than halves the SliderQuant perplexity; at 3 bits it restores the diverged GPTQ baseline to a meaningful perplexity and accuracy.

The improvement extends from the solvers GPTQ, AWQ, and QuIP to SliderQuant, a state-of-the-art PTQ method, indicating that the objective-level modification is broadly applicable rather than tied to a particular solver. Generic Fisher matches Task Fisher while removing its dependence on a labeled per-task training split and reusing the calibration data, so we adopted it as the default estimator for the remaining experiments. Across solvers the regularizer improves accuracy where the baseline leaves room and otherwise stays within seed-level noise, so the objective-level modification helps task accuracy without degrading it. The smaller-magnitude AWQ gains reflect its restricted one-dimensional scaling search space, which limits how much the Fisher term can reshape the solution. Because the AWQ family is a subset of the unconstrained quantizer set, its Fisher-weighted optimum can be no better than the one the Hessian-level solvers minimize (we prove this in the appendix).

Table 2: Cross-model multiple-choice accuracy. Each cell is the Generic Fisher accuracy with its change Δ over the baseline in parentheses, colored green when Generic improves and red otherwise. The bottom row reports the mean Δ over the nine (bit width, solver) settings for each model.

Bits	Method	Llama-3.2-1B	Llama-3.2-3B	Llama-3-8B	Qwen3-8B	Llama-3-70B
INT2	AWQ	32.83 ±0.00 (+0.03)	32.62 ±0.02 (+0.05)	32.46 ±0.03 (−0.06)	33.13 ±0.04 (+0.30)	33.04 ±0.04 (−0.06)
	GPTQ	32.88 ±0.11 (−0.31)	33.21 ±0.16 (−0.05)	32.90 ±0.19 (−0.05)	32.92 ±0.20 (+0.16)	32.96 ±0.15 (−0.07)
	QuIP	33.93 ±0.16 (+0.49)	35.17 ±0.19 (+1.08)	35.52 ±0.17 (+0.63)	35.27 ±0.68 (+0.26)	37.18 ±0.59 (+1.28)
INT3	AWQ	47.43 ±0.06 (+0.28)	55.12 ±0.09 (−1.08)	59.65 ±0.06 (−0.26)	60.75 ±0.09 (+0.32)	67.53 ±0.04 (+0.23)
	GPTQ	35.06 ±0.04 (+1.92)	39.54 ±0.32 (+6.24)	38.14 ±0.17 (+5.02)	49.52 ±0.34 (−1.19)	32.77 ±0.04 (+0.12)
	QuIP	48.32 ±0.27 (+0.91)	55.50 ±0.22 (+0.94)	59.65 ±0.30 (+0.59)	49.37 ±3.14 (+0.35)	45.82 ±2.55 (+5.12)
INT4	AWQ	52.25 ±0.04 (+0.68)	58.69 ±0.10 (−0.14)	63.29 ±0.03 (+0.01)	62.75 ±0.02 (−0.06)	69.63 ±0.02 (+0.05)
	GPTQ	49.02 ±0.12 (+3.67)	56.25 ±0.12 (+1.75)	61.57 ±0.07 (+3.34)	61.81 ±0.08 (+0.25)	32.79 ±0.13 (−0.58)
	QuIP	51.31 ±0.11 (+0.21)	58.42 ±0.07 (+0.41)	62.75 ±0.14 (+0.48)	54.72 ±2.15 (+6.30)	48.43 ±2.54 (+2.61)
Avg Δ		+0.88	+1.02	+1.08	+0.74	+0.97

5.3 Generalization Across Five LLMs

We tested whether Generic Fisher generalizes beyond Llama-3-8B, across two model families and scales from 1B to 70B. Table 2 reports multiple-choice accuracy for Generic Fisher against the baseline. The corresponding cross-model perplexity is in the appendix.

Generic Fisher improves the mean MC accuracy on every model. On QuIP it improves accuracy in every cell, the most consistent of the three. On GPTQ the gains are large at 3 and 4 bits, reaching $+6.24$ points on Llama-3.2-3B at 3 bits, but absent at 2 bits, where the baseline has collapsed and the regularizer cannot recover it. On AWQ the gains are small but positive in most settings, with a few regressions. The largest regressions are GPTQ on Qwen3-8B at 3 bits (-1.19) and AWQ on Llama-3.2-3B at 3 bits (-1.08). Separately, GPTQ on Llama-3-70B remains near chance accuracy at 3 and 4 bits with or without the regularizer, because its baseline itself diverges.

Generic Fisher generalizes across both families and across an order of magnitude in scale, which supports its use as a default rather than a per-model choice. Beyond accuracy, the regularizer also reduces perplexity in most settings (see appendix). A plausible explanation is that Generic Fisher is estimated from a CE loss on general text, which is aligned with the language-modeling objective that perplexity measures, so emphasizing the CE-salient weights reinforces rather than competes with the reconstruction objective. The Llama-3-70B GPTQ failures stem from a PTQ-side instability that the regularizer cannot repair, which we discuss in the limitations.

5.4 Where the Improvement Comes From

To test whether the regularizer protects the channels the loss is most sensitive to, the mechanism we posit behind these gains, we quantized Llama-3-8B at 4 bits with the three solvers, with and without Generic Fisher, and measured each input channel's quantization error as its mean squared deviation from the FP16 weights, pooling all linear layers and grouping channels into equal-frequency bins of Fisher importance. Figure 3 shows the resulting per-bin quantization error. The three solvers differ in how the regularizer reshapes the error. GPTQ lowers the error across the whole importance range, keeping a roughly constant gap below the baseline. QuIP nearly coincides with the baseline over most of the range but improves the highest-importance channels, at the cost of a small increase on the lowest-importance ones, the reallocation the regularizer is designed to induce. AWQ shows this reallocation only at the extremes, raising the error on the lowest-importance channels and lowering it on the highest. This mirrors the accuracy gains in tables 1 and 2, where GPTQ shows the largest improvements and QuIP and AWQ smaller ones, tying the size of each solver's gains to how much the regularizer reshapes its error.

5.5 Computational Overhead

The regularizer adds no inference-time cost. It changes only the quantization objective, so the quantized model has the same architecture, bit-width, and weight format as the unregularized baseline, and both its inference latency and its memory footprint are unchanged. At quantization time, the only cost the regularizer adds is computing the Fisher diagonal, which requires a single forward-backward

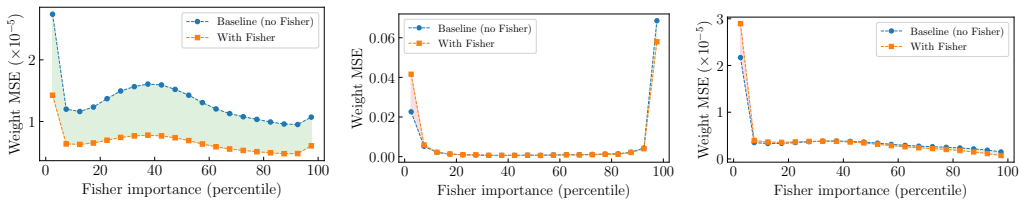

Figure 3: Quantization error versus Fisher importance on Llama-3-8B at 4 bits with Generic Fisher, for GPTQ, AWQ, and QuIP (one panel each). The y-axis is the per-channel Weight MSE and the x-axis orders channels by Fisher-importance percentile. Each curve is the per-bin mean over twenty equal-frequency bins for the unregularized baseline (blue) and the Generic Fisher model (orange). Green shading marks bins where the regularizer lowers the error and red shading where it raises it.

sweep over the calibration data and is then cached and reused across every (solver, bit-width, λ) configuration of a model. We compared this cost against one quantization run on a pure-compute basis that excludes model and data loading, with the full timings in the appendix. Computing the Fisher takes at most about one quantization run on the smallest model and substantially less on the larger ones, falling to 0.34–$0.66\times$ a single run on Llama-3-8B and 0.16–$0.53\times$ on Llama-3-70B. Because the Fisher is computed once and amortized over all configurations that reuse it, its overhead is a small fraction of the full quantization sweep.

6 Limitations

We approximate the Fisher information by its per-channel diagonal, consistent with the classical OBD saliency method (LeCun et al., 1989), which keeps the regularizer a simple diagonal loading but ignores cross-channel curvature, so it can underestimate directions whose sensitivity arises jointly across channels. A block-diagonal or K-FAC estimator (Martens & Grosse, 2015) would reflect the loss geometry more faithfully at higher construction and storage cost, which we leave to future work. Our theoretical justification is also local. The Fisher term is a second-order surrogate of the task-loss degradation under a stationarity assumption at the pretrained weights, a proxy-level statement rather than a global guarantee. Finally, because the regularizer acts on the objective rather than solver-side numerical stability, it cannot recover baselines whose failure is not objective-level, namely the most aggressive 2-bit greedy and grid-search solves, whose accuracy stays at chance, and the largest model under one solver, which diverges outright.

7 Conclusion

We presented an objective-level modification of layer-wise post-training quantization that augments the standard reconstruction loss with a Fisher-weighted regularizer. We instantiated it as a Task Fisher computed from a labeled task training split and a label-free Generic Fisher computed from the generic calibration data already used by PTQ. The resulting effective Hessian integrates without architectural change into any per-layer PTQ solver, and the Fisher term admits a local interpretation as a second-order Taylor approximation of the loss degradation. Empirically, Generic Fisher improves both perplexity and zero-shot accuracy over the unregularized baseline across model scales from 1B to 70B, multiple solvers, and 2–4 bit widths, in the large majority of settings, while using no labeled task training split and adding only a single forward-backward pass. A detailed analysis aligns the gains with the structural argument of the method, since the largest improvements occur in solvers that expose continuous Hessian-level handles and the smallest in grid-restricted solvers, and the regularizer composes with a state-of-the-art solver at small cost.

Natural directions for future work include pairing the regularizer with solver-side stabilization to address the 2-bit regime where the baseline collapses, and studying its interaction with quantization-aware fine-tuning.

AI USE STATEMENT

In this work, we used generative AI tools to propose and refine hypotheses, to design and provide feedback on the research methodology and experiments, to implement methods, to interpret results, to assist in the writing of proofs, and to assist with translation. We have not used generative AI tools to generate synthetic data sets, to develop theoretical models or conceptual frameworks, to formulate mathematical claims, to provide critical ingredients for proving mathematical claims, or to clean and reformat data sets, and qualitative and thematic data analysis is not applicable to this work. Additionally, we used generative AI tools for brainstorming, discovering research topics and identifying gaps, sourcing and searching for information, identifying relevant literature, summarizing and analyzing existing literature, drafting parts of the paper, editing the paper to improve readability, formatting references, and proposing a title and keywords.

We have reviewed all AI-assisted work. Hypotheses and experimental designs discussed with AI tools were checked against prior work through a manual literature survey, and all final research decisions were made by the authors. All AI-generated code was reviewed, tested, and verified for correctness by the authors. All mathematical claims were formulated by the authors, and every step of the AI-assisted proofs was verified by the authors. All interpretations of results were checked by the authors against the experimental data. Every paper identified or summarized with AI assistance was read in the original, and all references were verified to exist and to support the statements for which they are cited. All AI-drafted, AI-translated, and AI-edited text was revised and checked by the authors for technical accuracy. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure that the results in this paper are reproducible. The Fisher-regularized objective and the two Fisher estimators are defined in the Method section, and the integration of the regularizer into each PTQ solver, as well as the normalization and clipping of the Fisher diagonal, are detailed in the appendix. The experimental setup, including the models, solvers, calibration and Fisher data, hyperparameter grid, selection criterion for λ, and number of seeds, is described in the Experimental Setup subsection, and the exact definitions of the evaluation metrics are given in the appendix. All models, datasets, and the evaluation harness we use are publicly available. The full assumptions and complete proofs of the theoretical results, the per-task results, the cross-model perplexity, and the full timing measurements are also provided in the appendix. As supplementary material, we provide sample code for the Fisher computation and its integration into the PTQ solvers. The sample code is not a standalone executable pipeline, but it contains the core components of the method and is sufficient to understand and reimplement it together with the descriptions in the paper and appendix.

REFERENCES

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

Yamato Arai and Yuma Ichikawa. Quantization error propagation: Revisiting layer-wise post-training quantization. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M De Sa. Quip: 2-bit quantization of large language models with guarantees. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research (TMLR)*, 2024.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

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

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In *International Conference on Learning Representations (ICLR)*, 2023.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 2024. URL `https://zenodo.org/records/12608602`.

Ruihao Gong, Yang Yong, Shiqiao Gu, Yushi Huang, Chengtao Lv, Yunchen Zhang, Dacheng Tao, and Xianglong Liu. Llmc: Benchmarking large language model quantization with a versatile compression toolkit. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

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

Babak Hassibi and David Stork. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1992.

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. In *International Conference on Machine Learning (ICML)*, 2025.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017.

Sehoon Kim, Coleman Richard Charles Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. SqueezeLLM: Dense-and-sparse quantization. In *International Conference on Machine Learning (ICML)*, 2024.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 2017.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1989.

Amit LeVi, Raz Lapid, Rom Himelstein, Chaim Baskin, Ravid Shwartz Ziv, and Avi Mendelson. You had one job: Per-task quantization using llms' hidden representations. *arXiv preprint arXiv:2511.06516*, 2025.

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. In *International Conference on Learning Representations (ICLR)*, 2021.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. In *Conference on Machine Learning and Systems (MLSys)*, 2024.

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. In *International Conference on Learning Representations (ICLR)*, 2025.

Vladimir Malinovskii, Andrei Panferov, Ivan Ilin, Han Guo, Peter Richtárik, and Dan Alistarh. HIGGS: Pushing the limits of large language model quantization via the linearity theorem. In *Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL)*, 2025.

Mitch Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The penn treebank: Annotating predicate argument structure. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro*, 1994.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning (ICML)*, 2015.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*, 2017.

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning (ICML)*, 2020.

Yury Nahshan, Brian Chmiel, Chaim Baskin, Evgenii Zheltonozhskii, Ron Banner, Alex M Bronstein, and Avi Mendelson. Loss aware post-training quantization. *Machine Learning*, 2021.

Davide Paglieri, Saurabh Dash, Tim Rocktäschel, and Jack Parker-Holder. Outliers and calibration sets have diminishing effect on quantization of modern llms. *arXiv preprint arXiv:2405.20835*, 2024.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research (JMLR)*, 2020.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM (CACM)*, 2021.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Gao Peng, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Jack Sherman and Winifred J. Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Annals of Mathematical Statistics*, 1950.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip#: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *International Conference on Machine Learning (ICML)*, 2024.

Shigeng Wang, Chao Li, Yangyuxuan Kang, Jiawei Fan, Zhonghong Ou, and Anbang Yao. Sliderquant: Accurate post-training quantization for LLMs. In *International Conference on Learning Representations (ICLR)*, 2026.

Miles Williams and Nikolaos Aletras. On the impact of calibration data in post-training quantization and pruning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Miles Williams, George Chrysostomou, and Nikolaos Aletras. Self-calibration for language model quantization and pruning. In *Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL)*, 2025.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning (ICML)*, 2023.

Haiyang Xiao, Weiqing Li, Jinyue Guo, Guochao Jiang, Guohua Liu, and Yuewei Zhang. Faq: Mitigating quantization error via regenerating calibration data with family-aware quantization. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026.

Hanqi Xiao, Yi-Lin Sung, Elias Stengel-Eskin, and Mohit Bansal. Task-circuit quantization: Leveraging knowledge localization and interpretability for compression. In *Conference on Language Modeling (COLM)*, 2025.

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

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Haoyu Zhang, Shihao Zhang, Ian Colbert, and Rayan Saab. Provable post-training quantization: Theoretical analysis of optq and qronos. *arXiv preprint arXiv:2508.04853*, 2025.

Jiaqi Zhao, Ming Wang, Miao Zhang, Yuzhang Shang, Xuebo Liu, Yaowei Wang, Min Zhang, and Liqiang Nie. Benchmarking post-training quantization in llms: Comprehensive taxonomy, unified evaluation, and comparative analysis. *arXiv preprint arXiv:2502.13178*, 2025.

A Theoretical Results

This appendix collects the formal statements behind the theoretical interpretation of eq. (3), together with their proofs. They concern the per-layer quadratic objective and its effective Hessian, and they hold for both the Task Fisher and the Generic Fisher instantiations of F.

A.1 Setup and Notation

We consider a single linear layer with weight matrix $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ and output $y = Wx$ for $x \in \mathbb{R}^{d_{\text{in}}}$, and we write $\hat{W} \in \mathcal{Q}^{d_{\text{out}} \times d_{\text{in}}}$ for its quantized counterpart, where $\mathcal{Q}$ is a fixed scalar codebook, and $\Delta = \hat{W} - W$ for the quantization error. The calibration input activations are stacked as $X = [x^{(1)}, \dots, x^{(N)}] \in \mathbb{R}^{d_{\text{in}} \times N}$ with second-moment matrix $\Sigma_x = XX^\top$, and the reconstruction objective is $\mathcal{L}_{\text{recon}}(\hat{W}) = \operatorname{tr}(\Delta \Sigma_x \Delta^\top)$ as in eq. (1). Fixing a loss $\mathcal{L}$ and the data distribution on which it is evaluated, we write $g = \partial \mathcal{L} / \partial y \in \mathbb{R}^{d_{\text{out}}}$ for the gradient at the layer output. Since $\nabla_W \mathcal{L} = g x^\top \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, the exact weight-space Fisher matrix is

$$\mathcal{F}_W = \mathbb{E}\big[\operatorname{vec}(\nabla_W \mathcal{L}) \operatorname{vec}(\nabla_W \mathcal{L})^\top\big] = \mathbb{E}\big[(xx^\top) \otimes (gg^\top)\big], \tag{8}$$

where $\operatorname{vec}(A)$ denotes the column-wise vectorization of a matrix A, obtained by stacking its columns into a single column vector, and $\otimes$ denotes the Kronecker product. For matrices $A = [a_{uv}]$ and B, the Kronecker product $A \otimes B$ is the block matrix whose (u, v)-th block is $a_{uv} B$. We write I_d for the $d \times d$ identity matrix. We write $\kappa(A) = \lambda_{\max}(A) / \lambda_{\min}(A)$ for the condition number of a symmetric

positive definite matrix A. Therefore $\mathcal{F}_W \in \mathbb{R}^{d_{\text{in}}d_{\text{out}} \times d_{\text{in}}d_{\text{out}}}$. Storing eq. (8) is infeasible, so we use the channelwise diagonal surrogate $F = \operatorname{diag}(f_1, \ldots, f_{d_{\text{in}}})$ with $f_j = \mathbb{E}[\|g\|_2^2 x_j^2]$ from eq. (2), which yields the Fisher term $\operatorname{tr}(\Delta F \Delta^\top) = \sum_{j=1}^{d_{\text{in}}} f_j \|\Delta_{:,j}\|_2^2$ and the effective Hessian $H_{\text{eff}} = \Sigma_x + \lambda F$ of eq. (4). We denote by $\mathcal{M}_{\text{AWQ}} = \{Q(WS)S^{-1} : S \in \mathcal{S}\}$ the family of per-channel rescaled quantizers explored by AWQ, where $\mathcal{S}$ is a family of diagonal scaling matrices, and for a positive definite matrix H and a coordinate q with scalar rounding residual ε_q, we write $\psi_q(H) = \varepsilon_q^2/(H^{-1})_{qq}$ for the greedy quadratic error contributed by that coordinate, the quantity that GPTQ- and QuIP-style solvers accumulate.

A.2 Generalization and Local Curvature

This subsection clarifies both the scope and the local justification of the proposed objective. We first show that standard reconstruction-based PTQ, isotropically damped quadratic objectives, and Fisher-regularized AWQ problems arise as special cases, and then relate the Fisher penalty to a local second-order surrogate for the increase in the loss that defines F.

Proposition 1 (Special cases)**.** *For the Fisher-regularized objective in eq.* (3)*, the following hold.* (i) *If $\lambda = 0$, the objective reduces to the reconstruction objective $\mathcal{L}_{\text{recon}}$ in eq.* (1)*.* (ii) *If $F = \gamma I$ for some $\gamma \geq 0$, the effective Hessian becomes $H_{\text{eff}} = \Sigma_x + \lambda\gamma I$, an isotropically damped quadratic proxy.* (iii) *If the feasible set is restricted to $\mathcal{M}_{\text{AWQ}}$, minimizing eq.* (3) *is a Fisher-regularized AWQ problem over the scaling variable S.*

Proof. For (i), setting $\lambda = 0$ in eq. (3) removes the Fisher term, leaving $\mathcal{L}_{\text{FR}}(\hat{W}) = \mathcal{L}_{\text{recon}}(\hat{W})$ for any solver's reconstruction objective. For (ii), substituting $F = \gamma I$ into eq. (4) gives $H_{\text{eff}} = \Sigma_x + \lambda\gamma I$, which acts as isotropic damping. For (iii), restricting the minimization to $\mathcal{M}_{\text{AWQ}}$ gives $\hat{W}^* \in \arg\min_{\hat{W} \in \mathcal{M}_{\text{AWQ}}} \operatorname{tr}((\hat{W} - W)H_{\text{eff}}(\hat{W} - W)^\top)$. Since every feasible point has the form $\hat{W} = Q(WS)S^{-1}$ for some $S \in \mathcal{S}$, this is exactly a Fisher-regularized AWQ-type problem over S. □

Proposition 2 (Second-order loss-curvature surrogate)**.** *Assume the expected loss $\mathcal{L}(W)$ that defines F is twice continuously differentiable and stationary at the pretrained weights, $\nabla\mathcal{L}(W) = 0$. Then for any perturbation Δ with sufficiently small norm, we have*

$$\begin{aligned}\mathcal{L}(W + \Delta) - \mathcal{L}(W) = \tfrac{1}{2} \operatorname{vec}(\Delta)^\top \nabla^2\mathcal{L}(W) \operatorname{vec}(\Delta) \\ + o(\|\Delta\|_F^2).\end{aligned} \tag{9}$$

Replacing $\nabla^2\mathcal{L}(W)$ by the Fisher matrix in eq. (8) *and then by the row-shared diagonal surrogate $I_{d_{\text{out}}} \otimes F$ yields $\operatorname{vec}(\Delta)^\top (I_{d_{\text{out}}} \otimes F) \operatorname{vec}(\Delta) = \operatorname{tr}(\Delta F \Delta^\top)$, so the Fisher term is a second-order surrogate of the degradation of $\mathcal{L}$.*

Proof. Identify the weight space $\mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ with $\mathbb{R}^{d_{\text{out}}d_{\text{in}}}$ via vec. Since $\mathcal{L}$ is twice continuously differentiable and $\nabla\mathcal{L}(W) = 0$, the Taylor expansion at W gives eq. (9) because $\|\operatorname{vec}(\Delta)\|_2 = \|\Delta\|_F$. Writing the rows of Δ as $\Delta_{1,:}, \ldots, \Delta_{d_{\text{out}},:} \in \mathbb{R}^{1 \times d_{\text{in}}}$, the row-wise vec gives $\operatorname{vec}(\Delta)^\top (I_{d_{\text{out}}} \otimes F) \operatorname{vec}(\Delta) = \sum_{r=1}^{d_{\text{out}}} \Delta_{r,:} F \Delta_{r,:}^\top = \operatorname{tr}(\Delta F \Delta^\top)$. Substituting this row-shared diagonal surrogate for the Hessian in the second-order term proves the claim. □

Remark 1. Proposition 1 shows that the proposed method is best understood as a metric generalization of standard PTQ rather than a fundamentally different optimization problem, and proposition 2 is a local curvature justification, not a global guarantee about downstream task loss, since the stationarity assumption need not hold in general. The stationarity assumption is exact for the pretraining next-token-prediction (NTP) cross-entropy used by the full-sequence Generic Fisher and approximate, with bounded gradient norm, for Task Fisher and the last-token Generic Fisher. This viewpoint connects the regularizer to the classical OBD and OBS family of second-order saliency methods (LeCun et al., 1989; Hassibi & Stork, 1992).

A.3 Diagonal Loading and QuIP Compatibility

This subsection examines how the Fisher term modifies QuIP-style quadratic quantization and how it interacts with incoherence preprocessing. We first show that diagonal Fisher loading never decreases the greedy quadratic error of a coordinate, with an exact expression for rank-one loading, and then establish the orthogonal invariance required for compatibility with QuIP-style rotations.

Lemma 1 (Diagonal loading monotonicity). *Fix a positive definite matrix $H \in \mathbb{R}^{d\times d}$ and let D be any diagonal positive semi-definite (PSD) matrix, written $D \succeq 0$. Then $\psi_q(H+D) \geq \psi_q(H)$ holds for every coordinate q.*

Proof. Since $D \succeq 0$ and $H \succ 0$, we have $H + D \succeq H \succ 0$. Inversion reverses the Loewner order on positive definite matrices, so $(H+D)^{-1} \preceq H^{-1}$ and hence $((H+D)^{-1})_{qq} \leq (H^{-1})_{qq}$. Both diagonal entries are positive because H and $H+D$ are positive definite, so $\psi_q(H+D) = \varepsilon_q^2/((H+D)^{-1})_{qq} \geq \varepsilon_q^2/(H^{-1})_{qq} = \psi_q(H)$. □

Corollary 1 (Rank-one loading). *Under the setup of lemma 1, if $D = \alpha\, e_q e_q^\top$ for some $\alpha \geq 0$ and the q-th standard basis vector e_q, then we have*

$$\begin{aligned} \big((H + \alpha e_q e_q^\top)^{-1}\big)_{qq} &= \frac{(H^{-1})_{qq}}{1 + \alpha (H^{-1})_{qq}}, \\ \psi_q(H + \alpha e_q e_q^\top) &= \psi_q(H) + \alpha \varepsilon_q^2. \end{aligned} \tag{10}$$

Proof. Applying the Sherman–Morrison formula (Sherman & Morrison, 1950) to $H + \alpha e_q e_q^\top$ gives $(H+\alpha e_q e_q^\top)^{-1} = H^{-1} - \alpha H^{-1} e_q e_q^\top H^{-1}/(1+\alpha\, e_q^\top H^{-1} e_q)$. Since $e_q^\top H^{-1} e_q = (H^{-1})_{qq}$, taking the (q,q) entry yields the first identity in eq. (10). Substituting it into ψ_q gives $\psi_q(H + \alpha e_q e_q^\top) = \varepsilon_q^2(1 + \alpha(H^{-1})_{qq})/(H^{-1})_{qq} = \psi_q(H) + \alpha\varepsilon_q^2$. □

Proposition 3 (Orthogonal invariance). *Fix orthogonal matrices $U \in \mathbb{R}^{d_{\text{out}}\times d_{\text{out}}}$ and $V \in \mathbb{R}^{d_{\text{in}}\times d_{\text{in}}}$, and define the rotated quantities $\tilde{W} = UWV^\top$, $\tilde{\hat{W}} = U\hat{W}V^\top$, and $\tilde{H}_{\text{eff}} = VH_{\text{eff}}V^\top$. Then we have $\operatorname{tr}(\Delta H_{\text{eff}} \Delta^\top) = \operatorname{tr}(\tilde{\Delta}\, \tilde{H}_{\text{eff}}\, \tilde{\Delta}^\top)$ with $\tilde{\Delta} = \tilde{\hat{W}} - \tilde{W} = U\Delta V^\top$.*

Proof. With $\tilde{\Delta} = U\Delta V^\top$ we have $\operatorname{tr}(\tilde{\Delta}\, \tilde{H}_{\text{eff}}\, \tilde{\Delta}^\top) = \operatorname{tr}(U\Delta V^\top V H_{\text{eff}} V^\top V \Delta^\top U^\top) = \operatorname{tr}(U\Delta H_{\text{eff}} \Delta^\top U^\top) = \operatorname{tr}(\Delta H_{\text{eff}} \Delta^\top)$, using $V^\top V = I$, $U^\top U = I$, and the cyclic invariance of the trace. □

Remark 2. Lemma 1 and corollary 1 show that the Fisher term acts as anisotropic diagonal loading that selectively raises the greedy rounding cost of task-important channels, and proposition 3 implies that an exact Fisher-QuIP method may first form H_{eff} and then apply QuIP-style incoherence preprocessing without changing the objective value, which is the formulation aligned with the QuIP theory for PSD quadratic proxies (Chee et al., 2023). Injecting the diagonal Fisher after partial preprocessing, as we do in practice, is an approximation of this exact placement.

A.4 Restricted Families and Conditioning

This subsection characterizes the effects of restricting the feasible quantization family and adding Fisher-weighted curvature. We first show that optimization over an AWQ-type restricted family cannot outperform optimization over the unconstrained quantizer set under the same quadratic proxy, and then derive spectral bounds that quantify how the magnitude and anisotropy of the Fisher term affect the conditioning of the effective Hessian.

Proposition 4 (Restricted-family lower bound). *Fix any PSD matrix $H \in \mathbb{R}^{d_{\text{in}}\times d_{\text{in}}}$. If $\mathcal{M}_{\text{AWQ}} \subseteq \mathcal{Q}^{d_{\text{out}}\times d_{\text{in}}}$, then we have*

$$\begin{aligned} &\inf_{\hat{W}\in\mathcal{M}_{\text{AWQ}}} \operatorname{tr}\big((\hat{W}-W)H(\hat{W}-W)^\top\big) \\ &\qquad \geq \inf_{\hat{W}\in\mathcal{Q}^{d_{\text{out}}\times d_{\text{in}}}} \operatorname{tr}\big((\hat{W}-W)H(\hat{W}-W)^\top\big). \end{aligned} \tag{11}$$

Proof. Because $\mathcal{M}_{\mathrm{AWQ}} \subseteq \mathcal{Q}^{d_{\mathrm{out}} \times d_{\mathrm{in}}}$, the feasible set on the left-hand side of eq. (11) is a subset of the feasible set on the right-hand side. The infimum over a smaller set cannot be smaller than the infimum over the larger set, which is eq. (11). □

Proposition 5 (Conditioning bound)**.** *Assume $f_j \in [f_{\min}, f_{\max}]$ for all j, with $0 \le f_{\min} \le f_{\max}$. Then the effective Hessian satisfies $\lambda_{\min}(H_{\mathrm{eff}}) \ge \lambda_{\min}(\Sigma_x) + \lambda f_{\min}$ and $\lambda_{\max}(H_{\mathrm{eff}}) \le \lambda_{\max}(\Sigma_x) + \lambda f_{\max}$. Consequently, if $\lambda_{\min}(\Sigma_x) + \lambda f_{\min} > 0$, then we have*

$$\kappa(H_{\mathrm{eff}}) \le \frac{\lambda_{\max}(\Sigma_x) + \lambda f_{\max}}{\lambda_{\min}(\Sigma_x) + \lambda f_{\min}}. \tag{12}$$

Proof. Since each $f_j \in [f_{\min}, f_{\max}]$, the diagonal matrix F satisfies $f_{\min} I \preceq F \preceq f_{\max} I$. Multiplying by $\lambda \ge 0$ and adding Σ_x gives $\Sigma_x + \lambda f_{\min} I \preceq H_{\mathrm{eff}} \preceq \Sigma_x + \lambda f_{\max} I$. By monotonicity of the extreme eigenvalues under the Loewner order, $\lambda_{\min}(H_{\mathrm{eff}}) \ge \lambda_{\min}(\Sigma_x) + \lambda f_{\min}$ and $\lambda_{\max}(H_{\mathrm{eff}}) \le \lambda_{\max}(\Sigma_x) + \lambda f_{\max}$. If $\lambda_{\min}(\Sigma_x) + \lambda f_{\min} > 0$ then H_{eff} is positive definite, and dividing the two bounds gives eq. (12). □

Remark 3. Proposition 5 gives a theoretical interpretation of Fisher clipping. Bounding f_j at c_F controls $f_{\max}$ and therefore limits both the anisotropy introduced by the Fisher term and the resulting upper bound on the condition number of the per-layer system. Finally, these are proxy-level statements. A separate line of work studies quantitative error bounds for OPTQ/GPTQ under regularized covariance matrices and relates them to conditioning, regularization, and feature ordering (Zhang et al., 2025), and our formulation suggests an anisotropic extension of that framework in which the isotropic damping term is replaced by λF, which we do not establish here.

B Implementation Details

All experiments build on the PTQ-Bench framework (Zhao et al., 2025). Our GPTQ, AWQ, and QuIP quantizers are implemented on top of its solver implementations, and our SliderQuant quantizer is implemented on top of the released SliderQuant code (Wang et al., 2026). We add the Fisher regularizer to each without altering the underlying optimization routine.

B.1 Solver Configurations

We ran every solver in the configuration recommended by its original work, quantizing weights only and keeping activations in FP16. All solvers used the same calibration set of 128 segments of 2048 tokens. GPTQ used a single quantization scale per output row without finer grouping, with activation ordering enabled, asymmetric quantization, and the Hessian dampened by 1% of its mean diagonal. QuIP used LDLQ rounding with the full incoherence preprocessing pipeline, namely the Hessian preprocessing, the diagonal rescaling, and the random orthogonal projection with its additional step enabled, together with the symmetric incoherence-based quantizer, the same 1% Hessian dampening, and again no finer grouping. AWQ used a group size of 128 with asymmetric quantization and a zero point, and a single pass of the scaling grid search. SliderQuant used asymmetric per-channel weight quantization with a zero point, together with its learnable weight clipping and learnable equivalent transformation. It optimized a sliding window of two consecutive layers with a stride of one layer and a scheduling window of four layers, under its default progressive schedule of two stages with a stage quantization ratio of 0.5. Each stage ran for 60 epochs in bfloat16 with a training batch size of 3, LoRA adapters of rank 4, and learning rates of 5×10^{-4} for the LoRA parameters and 10^{-2} for the clipping parameters, with no weight decay and no warmup. Because these recommended configurations differ in quantization granularity, absolute numbers are not directly comparable across solvers, and every comparison we report is between the baseline and a proposed estimator within a fixed solver and bit width.

B.2 Effective Metric Construction

We describe the practical estimator used in all experiments, whose guiding principle is to preserve the PSD quadratic proxy while reusing the original optimization routines of GPTQ (Frantar et al., 2023), QuIP (Chee et al., 2023), and AWQ (Lin et al., 2024). We first estimate the channelwise Fisher surrogate from a small sample of the Fisher data under a surrogate loss, which are the task

Algorithm 1 Fisher diagonal construction

Require: quantized layer set $\mathcal{I}$, Fisher data $\mathcal{D}_F$, surrogate loss, cap c_F
Ensure: normalized Fisher diagonals $\{F_{\text{final},\ell}\}_{\ell\in\mathcal{I}}$
$\hat{f}_{\ell,j} \leftarrow 0$ for all $\ell \in \mathcal{I}$ and input channels j
for sample $s \in \mathcal{D}_F$ **do**
  run a forward pass, compute the surrogate loss, and backpropagate once to obtain the input activation x_ℓ and output gradient g_ℓ at each layer ℓ
  for all $\ell \in \mathcal{I}$ and input channels j of layer ℓ **do**
    $\hat{f}_{\ell,j} \leftarrow \hat{f}_{\ell,j} + \|g_\ell\|_2^2\, x_{\ell,j}^2$
  end for
end for
for $\ell \in \mathcal{I}$ **do**
  normalize and clip $\hat{f}_{\ell,j}$ by eqs. (13) and (14) to form $F_{\text{final},\ell}$
end for
return $\{F_{\text{final},\ell}\}_{\ell\in\mathcal{I}}$

training split and the task loss for Task Fisher and the generic calibration data and the cross-entropy loss for Generic Fisher. Writing $(x^{(m)}, g^{(m)})$ for the activation and output-gradient pair of the m-th of n_F samples, we estimate

$$\hat{f}_j = \frac{1}{n_F}\sum_{m=1}^{n_F} \|g^{(m)}\|_2^2\,(x_j^{(m)})^2, \qquad \hat{F} = \operatorname{diag}(\hat{f}_1, \ldots, \hat{f}_{d_{\text{in}}}). \tag{13}$$

We then normalize, clip, and re-normalize the Fisher scores within each layer, which is what makes λ a scale-invariant, dimensionless hyperparameter,

$$\begin{aligned} \tilde{f}_j &= \frac{\hat{f}_j}{\bar{f}}, \qquad \tilde{f}_j^{\text{clip}} = \min\{\tilde{f}_j, c_F\}, \\ f_j^{\text{final}} &= \frac{\tilde{f}_j^{\text{clip}}}{\bar{f}^{\text{clip}}}, \qquad F_{\text{final}} = \operatorname{diag}(f_1^{\text{final}}, \ldots, f_{d_{\text{in}}}^{\text{final}}), \end{aligned} \tag{14}$$

where $\bar{f} = \frac{1}{d_{\text{in}}}\sum_j \hat{f}_j$ and $\bar{f}^{\text{clip}} = \frac{1}{d_{\text{in}}}\sum_j \tilde{f}_j^{\text{clip}}$ are the pre- and post-clipping layer means. The normalized Fisher diagonal F_{final} is the only quantity we precompute, and we cache it per model and reuse it across every solver, bit width, and λ. Solvers that expose a per-layer second moment Σ_x augment it with this Fisher diagonal to form the practical effective metric

$$\hat{H}_{\text{eff}} = \Sigma_x + \lambda\bar{h}\, F_{\text{final}}, \qquad \bar{h} = \tfrac{1}{d_{\text{in}}}\operatorname{tr}(\Sigma_x), \tag{15}$$

where the factor $\bar{h}$ places the loading on the same scale as the reconstruction curvature. Solvers that do not expose Σ_x instead add the Fisher-weighted penalty to their own objective, as detailed below. The Fisher construction is summarized in algorithm 1, and its integration into each solver in algorithm 2.

B.3 Solver-Specific Integration

The proposed method is solver-agnostic at the quadratic-proxy level. For any layer-wise solver $\mathcal{A}$ we run its original routine unchanged and inject the cached Fisher diagonal F_{final} from algorithm 1, regardless of whether it comes from Task Fisher or Generic Fisher. Write $\mathcal{A}(W; H)$ for the quantized weights returned by solver $\mathcal{A}$ under the per-layer quadratic metric H, so that the standard solver is $\mathcal{A}(W; \Sigma_x)$. Solvers that expose Σ_x form the effective metric $\hat{H}_{\text{eff}}$ of eq. (15) from their own Σ_x and run $\mathcal{A}(W; \hat{H}_{\text{eff}})$, while solvers that do not add the Fisher-weighted penalty to their objective. The plug-in is summarized in algorithm 2, and the only solver-specific choice is where the injection is applied, which we detail below.

GPTQ. The GPTQ update depends on the Hessian only through H and H^{-1}, so replacing H with $\hat{H}_{\text{eff}}$ from eq. (15) leaves the closed-form solution unchanged in form. In practice we add the diagonal after the calibration statistics are accumulated and before the column-wise quantization loop.

Algorithm 2 Fisher-regularized PTQ with solver plug-in

Require: layer weights $\{W_\ell\}_{\ell\in\mathcal{I}}$, normalized Fisher diagonals $\{F_{\mathrm{final},\ell}\}_{\ell\in\mathcal{I}}$, strength λ, layer-wise solver $\mathcal{A}$
Ensure: quantized weights $\{\hat{W}_\ell\}_{\ell\in\mathcal{I}}$
for $\ell \in \mathcal{I}$ **do**
  if $\mathcal{A}$ exposes the per-layer second moment $\Sigma_{x,\ell}$ **then**
    form $\hat{H}_{\mathrm{eff},\ell} \leftarrow \Sigma_{x,\ell} + \lambda\bar{h}\, F_{\mathrm{final},\ell}$ by eq. (15)
    $\hat{W}_\ell \leftarrow \mathcal{A}(W_\ell;\, \hat{H}_{\mathrm{eff},\ell})$ {Σ_x replaced by $\hat{H}_{\mathrm{eff}}$ (GPTQ, QuIP)}
  else
    run $\mathcal{A}$ on W_ℓ with the Fisher-weighted penalty added to its objective to obtain $\hat{W}_\ell$ {loss level (AWQ, SliderQuant)}
  end if
end for
return $\{\hat{W}_\ell\}_{\ell\in\mathcal{I}}$

QuIP. QuIP applies a random orthogonal preconditioning to H before its greedy update, and we add the Fisher loading before this preconditioning, so the subsequent rotation acts on $\hat{H}_{\mathrm{eff}}$. This placement preserves the QuIP analysis stated above.

AWQ. AWQ does not expose H explicitly, but instead grid-searches a per-channel rescaling S that minimizes a layer reconstruction loss. We keep the same scaling family $\hat{W}(S) = Q(WS)S^{-1}$ and replace the grid-search score by the Fisher-augmented criterion

$$\mathcal{L}_{\mathrm{AWQ}}(S) = \mathcal{L}_{\mathrm{recon}}(S) + \gamma_{\mathrm{norm}}\,\lambda\,\mathcal{L}_F(S), \tag{16}$$

where $\mathcal{L}_{\mathrm{recon}}(S)$ is the standard reconstruction loss, $\mathcal{L}_F(S)$ is the Fisher-weighted quantization-error term, and $\gamma_{\mathrm{norm}} > 0$ is an adaptive normalization computed within the current block that places the two terms on the same scale across grid points. This integration is at the loss level rather than the Hessian level, which structurally limits how much the regularizer can reshape the solution.

SliderQuant. SliderQuant also does not expose Σ_x, since it learns the quantization parameters of a sliding window of consecutive layers by gradient descent on the block-output reconstruction loss $\mathcal{L}_{\mathrm{block}}$. We integrate the regularizer at the loss level, adding to $\mathcal{L}_{\mathrm{block}}$ the Fisher-weighted penalty $\mathcal{L}_F$ of the form $\sum_j f_j \|\hat{W}_{:,j} - W_{:,j}\|_2^2$ evaluated on the current quantized weights of each linear module in the window and averaged over those modules. We weight this penalty by λ times an adaptive factor $\gamma_{\mathrm{norm}} = \mathcal{L}_{\mathrm{block}}/\mathcal{L}_F$ recomputed from the detached loss values at each optimization step, which keeps it on the same scale as the block loss, and we backpropagate the reconstruction and Fisher terms jointly. This is the gradient-based counterpart of the AWQ integration, applied over a block of layers rather than a per-channel scaling family.

C Evaluation Details

We evaluated every quantized model with a fixed protocol shared across solvers and bit widths.

C.1 Zero-Shot Task Metrics

We ran all zero-shot tasks without in-context examples through the LM evaluation harness (Gao et al., 2024). For the six multiple-choice tasks, ARC-easy, ARC-challenge, HellaSwag, PIQA, RACE, and WinoGrande, we report accuracy, scoring each option by its length-summed log-likelihood and selecting the highest-scoring one. For LAMBADA we report last-word prediction accuracy. For GSM8K we report the exact-match accuracy of the final answer under the harness flexible-extraction filter, and for TriviaQA we report the exact-match accuracy after whitespace normalization. All task scores are expressed as percentages.

C.2 PERPLEXITY

We measured perplexity on the test splits of C4, WikiText-2, and PTB (Raffel et al., 2020; Merity et al., 2017; Marcus et al., 1994). We concatenated each corpus, split it into non-overlapping segments of $T = 2048$ tokens, and computed the token-level next-token-prediction cross-entropy on every segment. The reported perplexity is the exponential of the mean per-token cross-entropy averaged over all segments of a corpus.

The aggregate perplexity reported in the main results averages WikiText-2 and PTB only, holding out C4 because it is the calibration data and the criterion for selecting λ. The per-corpus C4 perplexity still appears in the per-dataset tables.

D ADDITIONAL EXPERIMENTS

D.1 PER-TASK ACCURACY AND PERPLEXITY ON LLAMA-3-8B UNDER C4 CALIBRATION

We report the per-task breakdown behind the MC accuracy, Non-MC accuracy, and perplexity averages of table 1 for Llama-3-8B under C4 calibration. Table 3 lists the MC tasks, table 4 the Non-MC tasks, and table 5 the per-corpus perplexity. Two trends hold across the two accuracy tables. First, the per-task gains track the aggregate and are largest for GPTQ, whose baseline is the most degraded, for example at INT4 raising its MC average from 58.23 to 61.48 and its Non-MC average from 30.30 to 40.31, while they are smaller for QuIP and near-flat for AWQ. Second, the Non-MC tasks degrade far more than the MC tasks at low bit widths, where several baselines collapse to near zero, namely every method at INT2 and GPTQ at INT3. The regularizer does not lift these collapsed baselines meaningfully, and instead recovers the most Non-MC accuracy where the baseline is degraded but not collapsed, as for QuIP at INT3 (30.84 to 34.27) and GPTQ at INT4. The breakdown shows that the averages in table 1 are not driven by a single task, and that the free-form generation tasks are the bit-sensitive regime where preserving the task loss matters most. Task and Generic Fisher give comparable per-task accuracy, so the label-free Generic estimator recovers the task-level gains without task supervision.

The perplexity breakdown in table 5 mirrors the accuracy trends. GPTQ shows the largest reductions, recovering the diverged INT3 baseline to roughly 105 and lowering the INT4 average from 12.05 to 10.23, while AWQ and QuIP improve by smaller margins. The reductions are consistent across WikiText-2 and PTB, and Task and Generic Fisher perform comparably. At INT2 perplexity stays diverged for AWQ and GPTQ, matching the accuracy floor, and only QuIP remains measurable.

D.2 CROSS-MODEL PERPLEXITY

We report perplexity averaged over WikiText-2 and PTB, holding out C4 because it is used for calibration and λ selection, for Generic Fisher on the four models besides Llama-3-8B in table 6. Llama-3-8B is reported separately in table 5. Each cell shows the Generic Fisher perplexity and its change over the baseline, so the sign and color summarize whether the regularizer helps on each model. Generic Fisher lowers perplexity in most settings, with the largest reductions where the baseline is most degraded. At INT2 AWQ and GPTQ stay diverged across models while QuIP remains measurable and improves, so the consistent gains concentrate at INT3 and INT4.

D.3 QUANTIZATION OVERHEAD

Table 7 reports the wall-clock time for computing the Generic Fisher against one quantization run of each solver, on a pure-compute basis that excludes model and data loading. The Fisher column times a single forward-backward sweep over the 1024 C4 calibration segments, and the solver columns time the quantization body only. Computing the Fisher costs at most about one quantization run on the smallest model and substantially less on the larger ones, and because it is computed once and shared across all (solver, bit-width, λ) configurations of a model its amortized overhead is small relative to a full quantization sweep.

Table 3: Per-task MC zero-shot accuracy on Llama-3-8B under C4 calibration, the breakdown behind the MC average in table 1. Within each (bit, method) block, bold marks the better of the baseline and a proposed estimator in each column. The two proposed estimators are not ranked against each other.

Bits	Method	Fisher	ARC-c	ARC-e	HellaS	PIQA	RACE	WinoG	Average
FP16	-	-	50.34	80.09	60.16	79.71	40.10	72.69	63.85
INT2	AWQ	-	21.42 ± 0.00	24.16 ± 0.00	25.56 ± 0.00	52.67 ± 0.00	21.91 ± 0.00	**49.41 ± 0.00**	32.52 ± 0.00
		Task	**21.72 ± 0.03**	**24.28 ± 0.01**	**25.59 ± 0.00**	**52.72 ± 0.00**	**22.01 ± 0.00**	49.38 ± 0.29	**32.62 ± 0.05**
		Generic	**21.67 ± 0.00**	24.12 ± 0.00	**25.58 ± 0.00**	52.50 ± 0.07	**22.14 ± 0.02**	48.75 ± 0.12	32.46 ± 0.03
	GPTQ	-	21.79 ± 0.53	25.34 ± 0.46	25.65 ± 0.04	**53.57 ± 0.53**	21.40 ± 0.70	49.91 ± 0.41	32.94 ± 0.29
		Task	**21.83 ± 0.64**	**25.46 ± 0.30**	**25.67 ± 0.08**	53.42 ± 0.31	**21.97 ± 0.21**	49.58 ± 0.37	**32.99 ± 0.11**
		Generic	**22.37 ± 0.31**	24.65 ± 0.22	25.60 ± 0.11	53.34 ± 0.42	21.34 ± 0.35	**50.09 ± 0.94**	32.90 ± 0.19
	QuIP	-	19.83 ± 0.45	**28.46 ± 0.89**	29.40 ± 0.43	**54.83 ± 0.49**	26.28 ± 0.43	50.58 ± 0.45	34.90 ± 0.42
		Task	**20.02 ± 0.38**	28.18 ± 0.65	**31.50 ± 0.47**	54.19 ± 0.38	**26.64 ± 0.26**	**51.46 ± 0.45**	**35.33 ± 0.18**
		Generic	**20.24 ± 0.51**	28.43 ± 0.45	**31.03 ± 0.23**	54.53 ± 0.58	**27.12 ± 0.31**	**51.79 ± 0.36**	**35.52 ± 0.17**
INT3	AWQ	-	**43.60 ± 0.00**	**74.92 ± 0.00**	54.33 ± 0.00	76.61 ± 0.00	**39.90 ± 0.00**	70.09 ± 0.00	**59.91 ± 0.00**
		Task	41.96 ± 0.02	73.75 ± 0.08	**54.75 ± 0.01**	76.56 ± 0.04	39.54 ± 0.11	**71.52 ± 0.04**	59.68 ± 0.02
		Generic	42.03 ± 0.28	73.56 ± 0.09	**54.74 ± 0.01**	**76.63 ± 0.12**	39.33 ± 0.08	**71.59 ± 0.12**	59.65 ± 0.06
	GPTQ	-	**21.16 ± 0.23**	25.82 ± 0.24	26.34 ± 0.09	53.32 ± 0.36	22.62 ± 0.36	49.47 ± 0.78	33.12 ± 0.08
		Task	20.09 ± 0.22	**36.50 ± 0.80**	**32.15 ± 0.38**	**58.69 ± 0.41**	**28.04 ± 0.42**	**51.79 ± 0.40**	**37.88 ± 0.13**
		Generic	19.98 ± 0.53	**36.30 ± 0.53**	**32.03 ± 0.35**	**59.26 ± 0.39**	**27.48 ± 0.69**	**53.78 ± 0.34**	**38.14 ± 0.17**
	QuIP	-	41.21 ± 0.40	72.91 ± 0.75	55.09 ± 0.16	76.00 ± 0.24	38.24 ± 0.48	70.92 ± 0.54	59.06 ± 0.33
		Task	**42.25 ± 0.56**	**73.96 ± 0.93**	**55.57 ± 0.16**	**76.84 ± 0.36**	**39.23 ± 0.32**	**72.12 ± 0.45**	**60.00 ± 0.27**
		Generic	**41.88 ± 1.12**	**74.13 ± 0.60**	**55.55 ± 0.26**	**76.41 ± 0.25**	**39.22 ± 0.19**	70.73 ± 0.16	**59.65 ± 0.30**
INT4	AWQ	-	**49.74 ± 0.00**	**79.84 ± 0.00**	59.28 ± 0.00	**78.56 ± 0.00**	39.04 ± 0.00	73.16 ± 0.00	63.27 ± 0.00
		Task	49.49 ± 0.00	78.70 ± 0.00	**59.32 ± 0.00**	78.43 ± 0.03	**39.62 ± 0.07**	**74.29 ± 0.16**	**63.31 ± 0.03**
		Generic	**49.74 ± 0.03**	78.60 ± 0.11	59.21 ± 0.02	78.49 ± 0.01	**39.46 ± 0.10**	**74.21 ± 0.12**	**63.29 ± 0.03**
	GPTQ	-	39.90 ± 1.73	71.52 ± 1.61	53.09 ± 1.40	75.84 ± 0.70	39.27 ± 0.21	69.77 ± 0.29	58.23 ± 0.86
		Task	**45.09 ± 0.38**	**77.43 ± 0.23**	**57.52 ± 0.03**	**77.37 ± 0.21**	**40.19 ± 0.27**	**71.27 ± 0.20**	**61.48 ± 0.14**
		Generic	**44.88 ± 0.34**	**77.63 ± 0.18**	**57.62 ± 0.06**	**77.33 ± 0.09**	**39.79 ± 0.30**	**72.19 ± 0.28**	**61.57 ± 0.07**
	QuIP	-	46.11 ± 0.63	77.67 ± 0.36	58.10 ± 0.28	78.51 ± 0.24	**40.11 ± 0.58**	73.15 ± 0.42	62.27 ± 0.32
		Task	**47.88 ± 0.54**	**77.99 ± 0.25**	**58.43 ± 0.10**	78.39 ± 0.23	39.79 ± 0.39	72.82 ± 0.37	**62.55 ± 0.14**
		Generic	**47.47 ± 0.42**	**78.66 ± 0.30**	**58.47 ± 0.06**	**78.73 ± 0.20**	39.98 ± 0.25	**73.20 ± 0.20**	**62.75 ± 0.14**

D.4 Loss-Form Comparison: Per-Task Breakdown

We study how the loss form used to build the Generic Fisher affects the estimator on Llama-3-8B, comparing the default last-token CE with a full-sequence NTP CE variant. Table 8 reports the per-task accuracy over the six MC tasks for the two forms and the $\lambda{=}0$ baseline.

The two forms track each other closely across all bit widths and solvers, with the last-token and full-sequence averages within a few tenths of a point at every setting. Both improve over the baseline wherever the baseline is weak, most clearly for GPTQ, whose average rises from 33.12 to 38.14 and 38.28 at INT3 and from 58.23 to 61.57 and 61.34 at INT4, and for QuIP, while for AWQ the already strong baseline leaves little room and both forms stay near it. Neither form wins consistently, and their per-setting differences have no clear direction.

The comparison shows that the Generic Fisher is insensitive to the loss form, so the channel importance it captures does not depend on whether the loss is placed on the last token or on the full sequence. This robustness lets us keep the simpler last-token CE as the default Generic Fisher without sacrificing accuracy, and it leaves the full-sequence variant as an equivalent fallback.

D.5 Task Fisher versus Task-Data Calibration

We isolate the effect of the Fisher regularizer from that of the calibration data by comparing task-data calibration alone against task-data calibration augmented with Task Fisher, under the task-matched protocol in which each metric is taken from the experiment calibrated on its own task. Table 9 reports the comparison over the six MC tasks on Llama-3-8B.

Adding Task Fisher improves the MC average in eight of the nine bit and method settings, with the only exception a marginal tie for AWQ at INT3. The gains are largest where the baseline is weakest, most clearly for GPTQ, whose average rises from 33.36 to 39.87 at INT3 and from 59.44 to 62.12 at INT4, and they stay positive for QuIP and for AWQ where the baseline is already strong.

Because both rows use the same task-matched calibration data, these gains isolate the contribution of the Fisher regularizer from that of the calibration choice. They show that matching the calibration

Table 4: Per-task Non-MC zero-shot accuracy on Llama-3-8B under C4 calibration, the breakdown behind the Non-MC average in table 1. Within each (bit, method) block, bold marks the better of the baseline and a proposed estimator in each column. The two proposed estimators are not ranked against each other.

Bits	Method	Fisher	GSM8K	Lamb-s	TriviaQA	Average
FP16	-	-	15.09	69.20	63.50	49.26
INT2	AWQ	-	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**
		Task	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**
		Generic	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**
	GPTQ	-	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
		Task	**0.02 ± 0.02**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.01 ± 0.00**
		Generic	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**
	QuIP	-	0.89 ± 0.20	2.98 ± 0.58	0.01 ± 0.00	1.30 ± 0.23
		Task	**1.00 ± 0.05**	**3.92 ± 0.41**	**0.04 ± 0.01**	**1.65 ± 0.12**
		Generic	**1.26 ± 0.19**	**4.22 ± 0.56**	**0.01 ± 0.00**	**1.83 ± 0.20**
INT3	AWQ	-	5.76 ± 0.00	58.57 ± 0.00	45.71 ± 0.00	36.68 ± 0.00
		Task	**9.75 ± 0.03**	**58.66 ± 0.00**	**46.48 ± 0.03**	**38.30 ± 0.02**
		Generic	**8.58 ± 0.21**	**58.82 ± 0.14**	**45.73 ± 0.20**	**37.71 ± 0.08**
	GPTQ	-	0.47 ± 0.18	0.01 ± 0.01	0.00 ± 0.00	0.16 ± 0.06
		Task	**1.03 ± 0.09**	**5.84 ± 0.48**	**0.10 ± 0.04**	**2.32 ± 0.17**
		Generic	**0.92 ± 0.12**	**5.03 ± 0.62**	**0.09 ± 0.06**	**2.02 ± 0.23**
	QuIP	-	7.55 ± 0.91	59.87 ± 0.86	25.09 ± 2.37	30.84 ± 1.19
		Task	**8.57 ± 1.32**	**60.64 ± 0.84**	**33.60 ± 3.22**	**34.27 ± 0.71**
		Generic	6.75 ± 0.68	**60.18 ± 0.48**	**30.07 ± 2.91**	**32.33 ± 1.14**
INT4	AWQ	-	10.92 ± 0.00	**68.70 ± 0.00**	59.37 ± 0.00	46.33 ± 0.00
		Task	**13.27 ± 0.00**	67.49 ± 0.00	**59.37 ± 0.00**	**46.71 ± 0.00**
		Generic	**12.12 ± 0.16**	67.75 ± 0.03	**59.52 ± 0.09**	**46.46 ± 0.04**
	GPTQ	-	6.76 ± 0.66	53.29 ± 1.87	30.85 ± 3.74	30.30 ± 1.76
		Task	**9.73 ± 0.45**	**60.99 ± 0.73**	**50.22 ± 1.25**	**40.31 ± 0.37**
		Generic	**8.54 ± 0.42**	**62.04 ± 0.67**	**48.51 ± 1.31**	**39.69 ± 0.34**
	QuIP	-	**15.27 ± 1.47**	65.31 ± 0.68	48.99 ± 1.78	43.19 ± 1.02
		Task	14.97 ± 0.99	**65.64 ± 0.60**	45.87 ± 2.10	42.16 ± 0.72
		Generic	12.24 ± 1.42	**65.68 ± 0.56**	**51.84 ± 1.11**	**43.25 ± 0.63**

data to the task is not sufficient on its own, and that the objective-level Fisher term adds accuracy on top of it across all three solvers. This supports our central claim that the improvement comes from the regularizer rather than from the data, and it mirrors the main results in that the largest gains appear for the most degraded baseline.

D.6 Overlap of Fisher Importance Across Tasks

A single generic estimator can match per-task estimators only if the weight directions the Fisher diagonal marks as important are largely shared across tasks, while any remaining benefit of a task-specific estimator must come from the directions on which the tasks disagree. We test this directly by comparing the cached Fisher diagonals themselves, without quantizing. For Llama-3-8B we load the per-channel Fisher diagonal computed for the six multiple-choice evaluation tasks, the three non-multiple-choice tasks, and the generic C4 estimator, all with 1024 samples at seed 42, pool the channels across all linear layers, and compare every pair with two measures. Cosine similarity captures agreement in the overall importance profile, which is dominated by the few channels of largest magnitude. To capture agreement on the critical set that the regularizer actually protects, we take each estimator's top 10% most important channels and measure how much these two sets coincide with the Jaccard index, the size of their intersection divided by the size of their union, which equals 1 when the sets are identical and 0 when they are disjoint. Because the multiple-choice and non-multiple-choice estimators also differ in the loss used to compute the Fisher, a comparison across these two groups reflects this loss-form difference as well as the difference between the tasks.

Figure 4 shows that the two measures tell a consistent two-part story. The importance profiles are highly aligned, with a mean cosine similarity of 0.94 between task pairs and 0.89 between the

Table 5: Per-corpus perplexity on Llama-3-8B under C4 calibration for the baseline and the proposed estimators across the PTQ solvers, the breakdown behind the perplexity average in table 1. The Average (except C4) column is the mean over the held-out corpora, excluding C4 because it is used to select λ. Perplexity of 1000 or above is shown as "–". Because perplexity is task-independent, the Task Fisher row uses the HellaSwag task Fisher as a representative instance. Within each (bit, method) block, bold marks the better of the baseline and a proposed estimator in each column. The two proposed estimators are not ranked against each other.

Bits	Method	Fisher	WikiText-2	PTB	C4	**Average (except C4)**
FP16	-	-	6.14	10.59	8.89	8.36
INT2	AWQ	-	–	–	–	–
		Task	–	–	–	–
		Generic	–	–	–	–
	GPTQ	-	–	–	–	–
		Task	–	–	–	–
		Generic	–	–	–	–
	QuIP	-	154.93 ± 29.37	183.13 ± 33.00	97.99 ± 12.66	169.03 ± 30.90
		Task	**88.92 ± 3.72**	**117.66 ± 6.72**	**61.68 ± 2.08**	**103.29 ± 4.41**
		Generic	**96.28 ± 5.93**	**128.88 ± 11.88**	**66.38 ± 1.09**	**112.58 ± 8.14**
INT3	AWQ	-	8.58 ± 0.00	13.90 ± 0.00	12.09 ± 0.00	11.24 ± 0.00
		Task	**8.17 ± 0.00**	**13.33 ± 0.00**	**11.57 ± 0.00**	**10.75 ± 0.00**
		Generic	**8.16 ± 0.00**	**13.36 ± 0.00**	**11.56 ± 0.00**	**10.76 ± 0.00**
	GPTQ	-	–	–	–	–
		Task	**81.58 ± 11.23**	**128.31 ± 11.36**	**74.44 ± 7.96**	**104.94 ± 11.24**
		Generic	**76.47 ± 12.41**	**142.20 ± 13.89**	**69.85 ± 7.78**	**109.34 ± 12.70**
	QuIP	-	8.49 ± 0.13	13.08 ± 0.23	11.67 ± 0.14	10.79 ± 0.18
		Task	**8.04 ± 0.07**	**12.74 ± 0.11**	**11.22 ± 0.06**	**10.39 ± 0.09**
		Generic	**8.13 ± 0.04**	**12.65 ± 0.06**	**11.29 ± 0.05**	**10.39 ± 0.05**
INT4	AWQ	-	6.57 ± 0.00	11.14 ± 0.00	9.43 ± 0.00	8.85 ± 0.00
		Task	**6.56 ± 0.00**	**11.07 ± 0.00**	**9.42 ± 0.00**	**8.81 ± 0.00**
		Generic	**6.55 ± 0.00**	**11.07 ± 0.00**	**9.42 ± 0.00**	**8.81 ± 0.00**
	GPTQ	-	9.21 ± 0.26	14.90 ± 0.60	12.44 ± 0.13	12.05 ± 0.37
		Task	**7.77 ± 0.05**	**12.70 ± 0.05**	**10.96 ± 0.03**	**10.23 ± 0.04**
		Generic	**7.74 ± 0.03**	**12.72 ± 0.05**	**11.00 ± 0.03**	**10.23 ± 0.04**
	QuIP	-	7.10 ± 0.11	11.56 ± 0.12	10.12 ± 0.13	9.33 ± 0.11
		Task	**6.96 ± 0.04**	**11.34 ± 0.02**	**9.94 ± 0.05**	**9.15 ± 0.03**
		Generic	**6.94 ± 0.04**	**11.39 ± 0.05**	**9.89 ± 0.05**	**9.16 ± 0.04**

Table 6: Cross-model perplexity for Generic Fisher, averaged over WikiText-2 and PTB with C4 held out because it is used for calibration and λ selection. Each cell is the Generic Fisher value (mean ± SEM) and its change Δ over the baseline, colored green when Generic Fisher improves and red otherwise. Perplexity of 1000 or above is shown as "–". When a cell reports a finite value its change is still shown even if the baseline exceeds this threshold, so a large negative change indicates recovery from a diverged baseline. Llama-3-8B is reported separately in table 5.

Bits	Method	Llama-3.2-1B	Llama-3.2-3B	Qwen3-8B	Llama-3-70B
INT2	AWQ	–	–	–	–
	GPTQ	–	–	–	–
	QuIP	–	226.27 ±26.20 (−293.29)	252.56 ±134.95 (−313.99)	62.66 ±6.42 (−5.62)
INT3	AWQ	23.31 ±0.06 (−0.44)	13.36 ±0.01 (−0.23)	15.42 ±0.03 (−0.31)	6.91 ±0.00 (−0.08)
	GPTQ	288.54 ±25.39 (−28683.61)	69.85 ±6.40 (−13333.04)	19.25 ±0.12 (−0.05)	–
	QuIP	18.64 ±0.18 (−1.57)	12.91 ±0.07 (−0.50)	63.80 ±26.20 (+22.74)	29.41 ±3.10 (−10.86)
INT4	AWQ	14.60 ±0.00 (−0.01)	10.82 ±0.00 (+0.01)	13.67 ±0.01 (+0.03)	5.79 ±0.00 (−0.02)
	GPTQ	19.35 ±0.12 (−1754.76)	12.94 ±0.09 (−5.94)	14.03 ±0.01 (−0.05)	–
	QuIP	14.78 ±0.04 (−0.33)	11.13 ±0.06 (−0.10)	40.81 ±10.11 (−16.54)	29.31 ±3.81 (+0.82)

generic estimator and each task, so the estimators share a dominant common backbone of important channels. This alignment holds even across the loss-form boundary, as the two free-form QA tasks GSM8K and TriviaQA stay above 0.95 with the multiple-choice tasks despite using a differ-

Table 7: Wall-clock time in seconds for computing the Generic Fisher versus one quantization run of each solver at INT4. Both are reported on a pure-compute basis that excludes model and data loading. The Fisher column is the forward-backward sweep over 1024 C4 segments, and the solver columns time the quantization body only. All timings use NVIDIA H100 80GB GPUs. †Measured on four GPUs, while all other models use one GPU.

Model	Fisher	GPTQ	AWQ	QuIP
Llama-3.2-1B	88.8 ±0.1	91.2 ±0.3	87.8 ±0.3	114.2 ±0.2
Llama-3.2-3B	169.0 ±0.5	213.7 ±0.5	208.4 ±0.5	271.9 ±0.3
Llama-3-8B	279.1 ±0.3	422.7 ±4.8	507.9 ±14.4	830.4 ±5.2
Qwen3-8B	321.2 ±0.1	376.7 ±2.4	474.1 ±2.7	727.9 ±1.8
Llama-3-70B†	1788.7 ±0.5	3381.6 ±6.7	3734.6 ±2.2	11441.8 ±30.5

Table 8: Generic Fisher loss-form comparison on Llama-3-8B under C4 calibration, the default last-token CE versus the full-sequence NTP CE variant, per task (accuracy). Within each (bit, method) block, bold marks the better of the baseline and a proposed form in each column. The two forms are not ranked against each other.

Bits	Method	Fisher	ARC-c	ARC-e	HellaS	PIQA	RACE	WinoG	Average
FP16	-	-	50.34	80.09	60.16	79.71	40.10	72.69	63.85
INT2	AWQ	-	21.42 ± 0.00	**24.16 ± 0.00**	25.56 ± 0.00	**52.67 ± 0.00**	21.91 ± 0.00	**49.41 ± 0.00**	**32.52 ± 0.00**
		Generic (last-token)	**21.67 ± 0.00**	24.12 ± 0.00	**25.58 ± 0.00**	52.50 ± 0.07	**22.14 ± 0.02**	48.75 ± 0.12	32.46 ± 0.03
		Generic (full-sequence)	**21.67 ± 0.00**	24.12 ± 0.00	**25.58 ± 0.00**	52.61 ± 0.00	**22.11 ± 0.00**	48.93 ± 0.00	32.50 ± 0.00
	GPTQ	-	21.79 ± 0.53	25.34 ± 0.46	**25.65 ± 0.04**	53.57 ± 0.53	21.40 ± 0.70	49.91 ± 0.41	32.94 ± 0.29
		Generic (last-token)	**22.37 ± 0.31**	24.65 ± 0.22	25.60 ± 0.11	53.34 ± 0.42	21.34 ± 0.35	**50.09 ± 0.94**	32.90 ± 0.19
		Generic (full-sequence)	**22.03 ± 0.61**	**25.50 ± 0.32**	25.65 ± 0.06	**53.83 ± 0.24**	**21.70 ± 0.38**	49.60 ± 0.36	**33.05 ± 0.14**
	QuIP	-	19.83 ± 0.45	**28.46 ± 0.89**	29.40 ± 0.43	**54.83 ± 0.49**	26.28 ± 0.43	50.58 ± 0.45	34.90 ± 0.42
		Generic (last-token)	**20.24 ± 0.51**	28.43 ± 0.45	**31.03 ± 0.23**	54.53 ± 0.58	**27.12 ± 0.31**	**51.79 ± 0.36**	**35.52 ± 0.17**
		Generic (full-sequence)	19.68 ± 0.38	28.23 ± 0.54	**31.60 ± 0.11**	54.73 ± 0.80	**26.55 ± 0.37**	**51.79 ± 0.22**	**35.43 ± 0.21**
INT3	AWQ	-	**43.60 ± 0.00**	**74.92 ± 0.00**	54.33 ± 0.00	76.61 ± 0.00	**39.90 ± 0.00**	70.09 ± 0.00	**59.91 ± 0.00**
		Generic (last-token)	42.03 ± 0.28	73.56 ± 0.09	**54.74 ± 0.01**	**76.63 ± 0.12**	39.33 ± 0.08	**71.59 ± 0.12**	59.65 ± 0.06
		Generic (full-sequence)	41.81 ± 0.08	73.44 ± 0.09	**54.76 ± 0.01**	**76.78 ± 0.02**	38.93 ± 0.11	**71.54 ± 0.02**	59.54 ± 0.01
	GPTQ	-	**21.16 ± 0.23**	25.82 ± 0.24	26.34 ± 0.09	53.32 ± 0.36	22.62 ± 0.36	49.47 ± 0.78	33.12 ± 0.08
		Generic (last-token)	19.98 ± 0.53	**36.30 ± 0.53**	**32.03 ± 0.35**	**59.26 ± 0.39**	**27.48 ± 0.69**	**53.78 ± 0.34**	**38.14 ± 0.17**
		Generic (full-sequence)	20.02 ± 0.72	**36.58 ± 0.74**	**32.62 ± 0.37**	**59.64 ± 0.69**	**28.15 ± 0.21**	**52.66 ± 0.36**	**38.28 ± 0.33**
	QuIP	-	41.21 ± 0.40	72.91 ± 0.75	55.09 ± 0.16	76.00 ± 0.24	38.24 ± 0.48	70.92 ± 0.54	59.06 ± 0.33
		Generic (last-token)	**41.88 ± 1.12**	**74.13 ± 0.60**	**55.55 ± 0.26**	**76.41 ± 0.25**	**39.22 ± 0.19**	70.73 ± 0.16	**59.65 ± 0.30**
		Generic (full-sequence)	41.13 ± 0.56	**73.79 ± 0.74**	**55.47 ± 0.34**	**76.53 ± 0.39**	**39.62 ± 0.57**	**71.13 ± 0.47**	**59.61 ± 0.31**
INT4	AWQ	-	49.74 ± 0.00	**79.84 ± 0.00**	**59.28 ± 0.00**	**78.56 ± 0.00**	39.04 ± 0.00	73.16 ± 0.00	63.27 ± 0.00
		Generic (last-token)	**49.74 ± 0.03**	78.60 ± 0.11	59.21 ± 0.02	78.49 ± 0.01	**39.46 ± 0.10**	**74.21 ± 0.12**	**63.29 ± 0.03**
		Generic (full-sequence)	**49.78 ± 0.02**	78.44 ± 0.03	59.20 ± 0.01	78.48 ± 0.01	**39.35 ± 0.07**	**74.29 ± 0.06**	63.26 ± 0.02
	GPTQ	-	39.90 ± 1.73	71.52 ± 1.61	53.09 ± 1.40	75.84 ± 0.70	39.27 ± 0.21	69.77 ± 0.29	58.23 ± 0.86
		Generic (last-token)	**44.88 ± 0.34**	**77.63 ± 0.18**	**57.62 ± 0.06**	**77.33 ± 0.09**	**39.79 ± 0.30**	**72.19 ± 0.28**	**61.57 ± 0.07**
		Generic (full-sequence)	**44.68 ± 0.40**	**76.76 ± 0.14**	**57.44 ± 0.10**	**77.33 ± 0.08**	**39.90 ± 0.17**	**71.95 ± 0.22**	**61.34 ± 0.08**
	QuIP	-	46.11 ± 0.63	77.67 ± 0.36	58.10 ± 0.28	78.51 ± 0.24	**40.11 ± 0.58**	73.15 ± 0.42	62.27 ± 0.32
		Generic (last-token)	**47.47 ± 0.42**	**78.66 ± 0.30**	**58.47 ± 0.06**	**78.73 ± 0.20**	39.98 ± 0.25	**73.20 ± 0.20**	**62.75 ± 0.14**
		Generic (full-sequence)	**48.19 ± 0.38**	**78.58 ± 0.31**	**58.49 ± 0.18**	**78.93 ± 0.28**	39.48 ± 0.36	72.87 ± 0.19	**62.76 ± 0.12**

ent Fisher loss, which indicates that the shared backbone is a genuine common core rather than an artifact of a shared loss form. The clearest exception is LAMBADA, whose last-word prediction stresses a more distinct set of channels and which sits near 0.81 with the other tasks and at 0.80 with the generic estimator. The critical sets, however, diverge, with a mean top-10% Jaccard overlap of only 0.33 between task pairs and 0.30 between the generic estimator and each task. This divergence is structured, with the two ARC variants overlapping at 0.80 and the two commonsense tasks HellaSwag and PIQA at 0.58, while less related pairs sit near 0.30–0.43 and LAMBADA overlaps least at 0.17–0.19. The generic estimator covers on average 0.46 of each task's top-10% channels, capturing the shared backbone while missing about half of the task-specific set. This is consistent with the results in the main text, where the generic estimator recovers most of the benefit because the important directions are largely shared, while a task-specific estimator retains a smaller edge from the critical channels the tasks do not share. The measures here are computed from a single seed.

D.7 HYPERPARAMETERS

The purpose of this study is to assess how the two hyperparameters of the method, the Fisher weight λ and the Fisher sample count n_F, affect performance for Generic Fisher on Llama-3-8B. Figures 5

Table 9: Task-matched calibration with and without Task Fisher on Llama-3-8B, per MC task (accuracy). Both rows use the same task-matched calibration, and the Fisher column marks whether the Task Fisher regularizer is added, with λ selected by C4 perplexity. Within each (bit, method) block, bold marks the higher-accuracy row. The Average column is the mean over the six MC tasks.

Bits	Method	Fisher	ARC-c	ARC-e	HellaS	PIQA	RACE	WinoG	Average
INT2	AWQ	-	**21.67 ± 0.00**	**24.28 ± 0.00**	25.60 ± 0.00	**52.61 ± 0.00**	21.91 ± 0.00	49.17 ± 0.00	32.54 ± 0.00
		Task	**21.67 ± 0.00**	24.18 ± 0.03	**25.65 ± 0.00**	**52.61 ± 0.05**	**21.95 ± 0.02**	**50.15 ± 0.43**	**32.70 ± 0.07**
	GPTQ	-	21.71 ± 0.41	24.12 ± 0.25	25.48 ± 0.08	53.26 ± 0.34	21.42 ± 0.45	**50.83 ± 0.46**	32.80 ± 0.15
		Task	**22.30 ± 0.66**	**25.09 ± 0.17**	**25.51 ± 0.07**	**53.50 ± 0.29**	**21.67 ± 0.25**	49.55 ± 0.57	**32.94 ± 0.09**
	QuIP	-	**20.68 ± 0.40**	25.98 ± 0.30	29.51 ± 0.32	**54.39 ± 0.67**	26.18 ± 0.48	51.14 ± 0.20	34.65 ± 0.29
		Task	20.58 ± 0.24	**26.20 ± 0.48**	**31.76 ± 0.14**	53.36 ± 0.44	**27.98 ± 0.35**	**52.41 ± 0.44**	**35.38 ± 0.11**
INT3	AWQ	-	**44.28 ± 0.00**	**75.76 ± 0.00**	54.27 ± 0.00	76.50 ± 0.00	**39.71 ± 0.00**	69.14 ± 0.00	**59.94 ± 0.00**
		Task	42.82 ± 0.02	75.33 ± 0.07	**54.89 ± 0.02**	**76.61 ± 0.00**	38.95 ± 0.00	**70.84 ± 0.07**	59.91 ± 0.02
	GPTQ	-	**21.06 ± 0.28**	25.55 ± 0.30	26.93 ± 0.10	53.75 ± 0.64	22.43 ± 0.58	50.43 ± 0.97	33.36 ± 0.13
		Task	20.10 ± 0.26	**38.60 ± 0.50**	**35.40 ± 0.67**	**61.41 ± 0.41**	**28.61 ± 0.27**	**55.11 ± 0.41**	**39.87 ± 0.10**
	QuIP	-	41.93 ± 0.51	73.41 ± 0.41	55.66 ± 0.52	76.77 ± 0.27	39.56 ± 0.45	**71.89 ± 0.48**	59.87 ± 0.29
		Task	**42.58 ± 0.48**	**75.17 ± 0.69**	**55.69 ± 0.16**	**77.07 ± 0.22**	**39.64 ± 0.29**	71.25 ± 0.62	**60.23 ± 0.14**
INT4	AWQ	-	**49.83 ± 0.00**	78.45 ± 0.00	**59.53 ± 0.00**	78.24 ± 0.00	40.48 ± 0.00	72.45 ± 0.00	63.16 ± 0.00
		Task	49.74 ± 0.00	**79.26 ± 0.01**	59.07 ± 0.02	**78.78 ± 0.02**	**40.82 ± 0.11**	**73.13 ± 0.10**	**63.47 ± 0.02**
	GPTQ	-	43.11 ± 0.49	73.43 ± 1.17	54.80 ± 0.60	75.71 ± 1.05	39.02 ± 0.09	70.59 ± 0.49	59.44 ± 0.29
		Task	**46.35 ± 0.33**	**77.72 ± 0.05**	**57.86 ± 0.09**	**78.06 ± 0.19**	**40.27 ± 0.40**	**72.45 ± 0.28**	**62.12 ± 0.08**
	QuIP	-	**47.22 ± 1.04**	77.77 ± 0.24	58.21 ± 0.13	78.26 ± 0.06	39.98 ± 0.25	**73.23 ± 0.24**	62.44 ± 0.18
		Task	47.20 ± 0.76	**78.09 ± 0.64**	**58.79 ± 0.24**	**78.30 ± 0.11**	**40.00 ± 0.20**	72.88 ± 0.37	**62.54 ± 0.10**

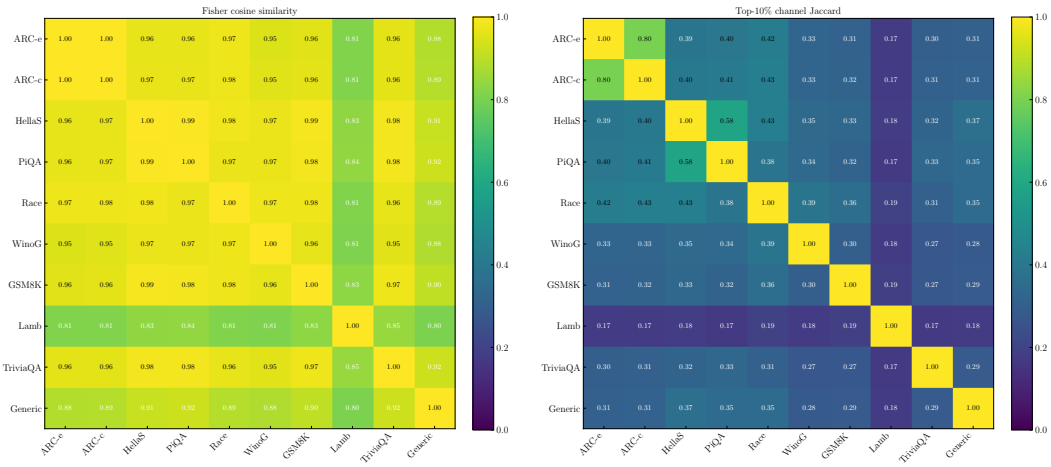

Figure 4: Overlap of Fisher importance across the six multiple-choice and three non-multiple-choice evaluation tasks and the generic C4 estimator on Llama-3-8B (1024 samples, seed 42). The left panel shows the cosine similarity of the per-channel Fisher diagonals, reflecting the overall importance profile. The right panel shows the overlap of the two estimators' top 10% most important channels, measured as the Jaccard index (shared channels divided by the union), where 1 means the two sets are identical and 0 means they share no channels. The profiles are highly aligned (left), including across the multiple-choice and free-form QA tasks, with LAMBADA the clearest exception, but the top-channel sets overlap far less and do so in a structured way, with related tasks such as the two ARC variants sharing more of their top channels (right).

and 6 report the accuracy averaged over all MC and Non-MC tasks for every PTQ solver and bit width, placing the unregularized baseline at $\lambda = 0$ and $n_F = 0$.

The method is robust to the Fisher weight λ. For most solver and bit-width combinations the averaged accuracy varies little across $\lambda \in \{0.1, 1.0, 2.0\}$, and any value in this range stays close to the oracle upper bound (λ selected per metric on the test set), so λ does not need careful tuning. The main exception is GPTQ at 3 and 4 bits, where accuracy rises with λ, consistent with the larger headroom of its degraded baseline. The averaged accuracy is also largely insensitive to the Fisher sample count n_F, since a modest number of samples already recovers most of the gain over the baseline and larger n_F changes the average only marginally, which supports estimating the Fisher

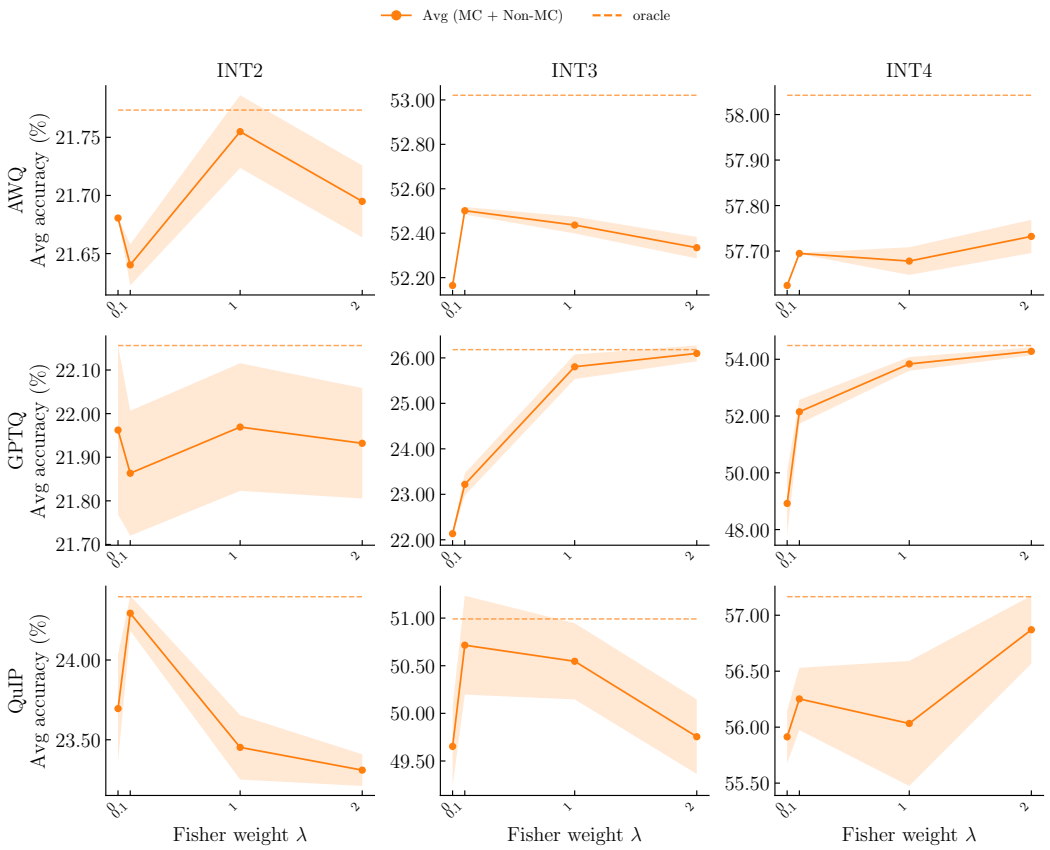

Figure 5: Sensitivity to the Fisher weight λ on Llama-3-8B (Generic Fisher, C4 calibration). Rows are PTQ solvers and columns are bit widths. Each curve is the accuracy averaged over all MC and Non-MC tasks as a function of λ, with $\lambda = 0$ the unregularized baseline. Shaded bands are the standard error over five seeds, and the dashed line is the oracle upper bound obtained by selecting, for each task, the λ that maximizes that task's own evaluation accuracy.

from a small sample. In many solver and bit-width settings, any swept value of λ or n_F yields higher averaged accuracy than the unregularized baseline, so the improvement does not hinge on a particular hyperparameter choice.

D.8 Accuracy and Perplexity on Llama-3-8B under WikiText-2 Calibration

To measure how much the calibration data affects the proposed estimators, we repeat the main comparison with WikiText-2 as the calibration data in place of C4. Table 10 reports the MC accuracy, Non-MC accuracy, and perplexity averages for the baseline, Task Fisher, and Generic Fisher, in the same format as table 1.

The results mirror the main C4-calibration table. The regularizer improves MC and Non-MC accuracy over the baseline in most settings, with the largest gains for GPTQ, whose baseline is the most degraded, for example raising its MC average from 33.23 to 38.57 at INT3 and from 58.87 to 61.89 at INT4. QuIP improves consistently, while AWQ improves by the smallest margin with gains mainly in Non-MC accuracy and perplexity and MC accuracy that stays roughly flat. Perplexity drops wherever it remains meaningful, and at INT2 both estimators stay collapsed except for QuIP, which remains measurable and improves.

Because switching the calibration corpus from C4 to WikiText-2 leaves the pattern and the size of the gains essentially unchanged, the benefit of the regularizer does not depend on the choice of calibration data. This is consistent with our design, in which the Fisher term acts on the quantization

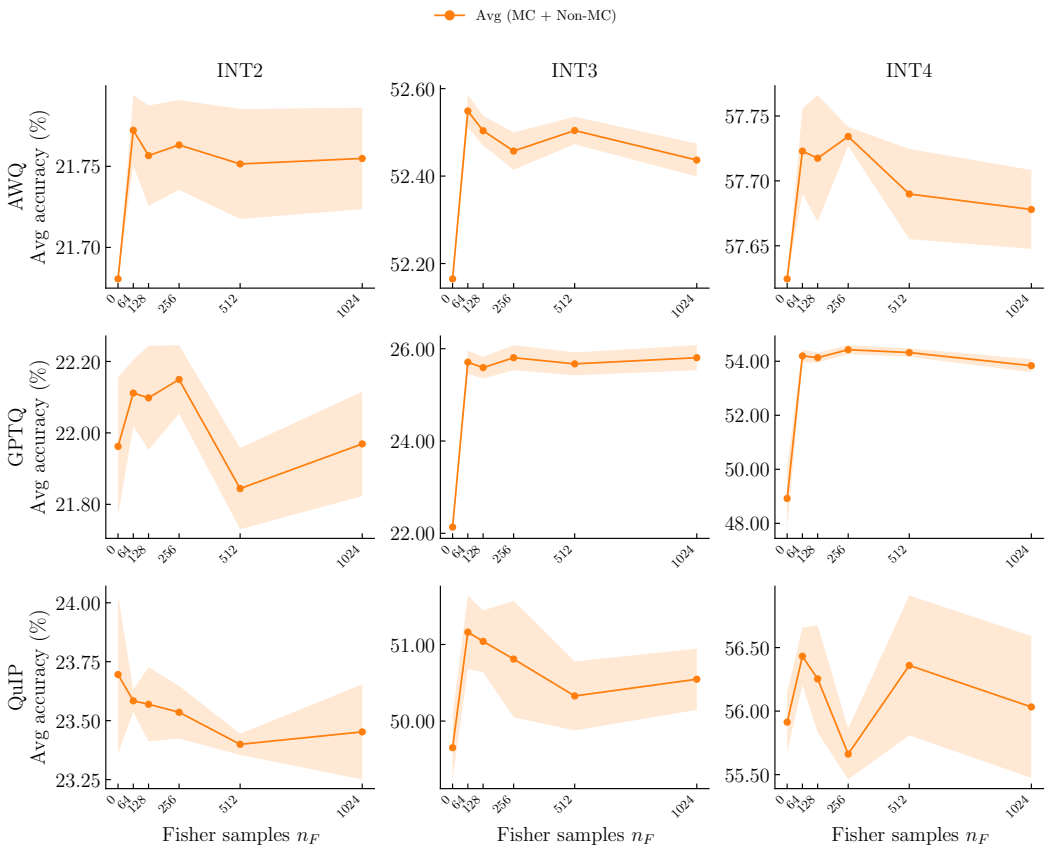

Figure 6: Sensitivity to the Fisher sample count n_F on Llama-3-8B (Generic Fisher, C4 calibration, $\lambda = 1.0$). Rows are PTQ solvers and columns are bit widths, as in fig. 5, with $n_F = 0$ the unregularized baseline. Shaded bands are the standard error over five seeds.

objective rather than on the calibration distribution, and it complements the task-data comparison in table 9 that isolates the same effect from the other direction.

Table 10: MC accuracy, Non-MC accuracy, and perplexity averages on Llama-3-8B under WikiText-2 calibration, in the same format as table 1. The MC and Non-MC columns average the six MC and three Non-MC tasks, and the PPL column is the mean over the held-out perplexity corpora, excluding WikiText-2 because it is used to select λ. For Task and Generic Fisher, λ is selected by WikiText-2 perplexity. For the Task Fisher row the MC and Non-MC columns use task-matched calibration. A dagger (†) marks a task-independent PPL, computed with the HellaSwag task Fisher. Perplexity of 1000 or above is shown as "–". Within each (bit, method) block, bold marks the better of the baseline and a proposed estimator in each column (higher for MC and Non-MC, lower for PPL). The two proposed estimators are not ranked against each other.

Bits	Method	Fisher	MC Avg (↑)	Non-MC Avg (↑)	PPL Avg (↓)
FP16	-	-	63.85	49.26	9.74
INT2	AWQ	-	32.40 ± 0.00	**0.00 ± 0.00**	–
		Task	**32.55 ± 0.06**	**0.00 ± 0.00**	–
		Generic	**32.68 ± 0.04**	**0.00 ± 0.00**	–
	GPTQ	-	**33.30 ± 0.12**	0.00 ± 0.00	–
		Task	32.97 ± 0.11	**0.01 ± 0.01**	–
		Generic	33.07 ± 0.15	**0.01 ± 0.01**	–
	QuIP	-	34.35 ± 0.11	1.28 ± 0.23	171.94 ± 13.70
		Task	**35.21 ± 0.12**	**1.57 ± 0.15**	**114.61 ± 6.96**†
		Generic	**34.71 ± 0.15**	**1.55 ± 0.21**	**116.33 ± 5.68**
INT3	AWQ	-	**60.16 ± 0.00**	36.99 ± 0.00	12.65 ± 0.00
		Task	59.34 ± 0.05	36.30 ± 0.03	**12.50 ± 0.00**†
		Generic	59.32 ± 0.06	**37.39 ± 0.08**	**12.51 ± 0.00**
	GPTQ	-	33.23 ± 0.15	0.21 ± 0.06	–
		Task	**37.86 ± 0.10**	**2.42 ± 0.17**	**101.83 ± 3.86**†
		Generic	**38.57 ± 0.17**	**2.82 ± 0.29**	**94.53 ± 5.58**
	QuIP	-	59.39 ± 0.44	33.25 ± 0.75	12.64 ± 0.12
		Task	**59.74 ± 0.29**	**34.05 ± 1.02**	**12.18 ± 0.08**†
		Generic	**59.58 ± 0.45**	**35.02 ± 0.34**	**12.27 ± 0.11**
INT4	AWQ	-	63.41 ± 0.00	45.94 ± 0.00	10.26 ± 0.00
		Task	**63.55 ± 0.03**	**46.20 ± 0.04**	**10.25 ± 0.00**†
		Generic	**63.81 ± 0.05**	**46.40 ± 0.06**	**10.25 ± 0.00**
	GPTQ	-	58.87 ± 0.22	33.04 ± 0.95	13.19 ± 0.10
		Task	**61.89 ± 0.10**	**39.82 ± 0.50**	**11.90 ± 0.05**†
		Generic	**61.55 ± 0.17**	**40.57 ± 0.37**	**11.83 ± 0.02**
	QuIP	-	61.59 ± 0.21	43.36 ± 0.98	10.92 ± 0.06
		Task	**62.26 ± 0.35**	**44.80 ± 0.61**	**10.67 ± 0.06**†
		Generic	**62.60 ± 0.20**	43.12 ± 0.41	**10.70 ± 0.04**