E-PMQ: EXPERT-GUIDED POST-MERGE QUANTIZATION WITH MERGED-WEIGHT ANCHORING

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-resource deployment constraints have made model quantization essential for deploying neural networks while preserving performance. Meanwhile, model merging has become an increasingly practical low-resource strategy for integrating multiple task- or domain-specialized experts into a single model without joint training or multi-model serving. Together, quantization and model merging enable an efficient low-resource deployment pipeline by integrating multiple experts into one low-bit model. We formulate this setting as *Post-Merge Quantization* (PMQ). We show that directly applying post-training quantization (PTQ) to a merged model is unreliable because two distinct deviations are coupled: the *quantization deviation* introduced by low-bit reconstruction and the *expert-relative merging deviation* inherited from model merging. To mitigate these deviations, we propose *E-PMQ*, an expert-guided PMQ framework that uses source expert weights to provide expert-guided output targets during layer-wise calibration, together with merged-weight anchoring to stabilize the calibration and preserve the integrated behavior of the merged model. Experiments on CLIP, FLAN-T5, and Llama show consistent gains over naive PMQ across vision and language, from 8-task CLIP-ViT-B/32 to 20-task CLIP-ViT-L/14 and up to 8B-scale LLMs.

Anonymous Code: `anonymous.4open.science/r/E-PMQ-demo-78B7`

1 INTRODUCTION

Low-resource deployment constraints have made model quantization essential for deploying neural networks while preserving performance. Low-bit post-training quantization (PTQ) is one of the most practical techniques for this setting, as it converts full-precision weights into low-bit representations using only a small calibration set and without expensive end-to-end retraining. Existing PTQ methods have achieved strong results for independently trained models, where the full-precision model is typically treated as a reliable reconstruction target during layer-wise quantization (Frantar et al., 2023; Lin et al., 2024; Xiao et al., 2023; Nagel et al., 2020; Li et al., 2021).

Model merging is also an increasingly practical low-resource strategy. Instead of jointly training a multi-task model or serving multiple experts, merging integrates several task- or domain-specialized models into a single model (Wortsman et al., 2022; Ilharco et al., 2023; Matena & Raffel, 2022; Yadav et al., 2023; Yu et al., 2024; Cheng et al., 2025). This makes merging attractive for resource-constrained adaptation and deployment: the resulting model can combine capabilities from multiple experts while avoiding multi-model serving. However, a merged model is not necessarily an independently optimized multi-task model. Since it is obtained through parameter composition, it may already deviate from the expert behaviors that merging aims to preserve.

These two low-resource techniques naturally meet in deployment: after experts are merged into a single model, the resulting model may still need to be quantized for low-bit inference. We formulate this setting as *Post-Merge Quantization* (PMQ), where the quantization target is a merged model rather than an independently trained model. This distinction is important because naive PMQ couples two distinct deviations. The first is the *quantization deviation* introduced by low-bit reconstruction. The second is the *expert-relative merging deviation* inherited from model merging. Directly applying ordinary PTQ methods such as GPTQ (Frantar et al., 2023) to a merged model only reconstructs the merged model itself, and therefore treats this potentially deviated model as the sole target. As a result, naive PMQ may preserve expert-relative merging deviations and further compound them

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

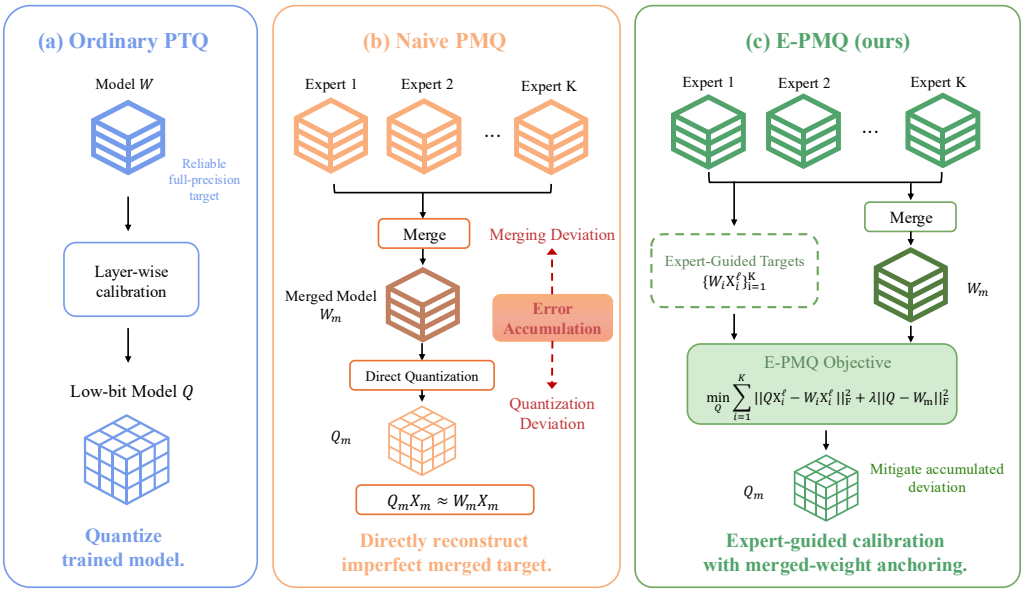


Figure 1: Overview of ordinary PTQ, naive PMQ, and E-PMQ. Ordinary PTQ quantizes a trained model by reconstructing a reliable full-precision target. Naive PMQ first merges multiple expert checkpoints and then directly quantizes the merged model, thereby reconstructing an imperfect merged target and suffering from accumulated merging and quantization deviation. E-PMQ instead uses expert-guided output targets during layer-wise calibration and anchors the quantized weight to the merged checkpoint W_m, turning post-merge quantization into expert-guided calibration with merged-weight anchoring.

with quantization deviation, making the standard merge-then-quantize pipeline unreliable, especially under aggressive low-bit settings.

To mitigate these deviations, we propose *E-PMQ*, an expert-guided PMQ framework with merged-weight anchoring. During layer-wise calibration, E-PMQ uses source expert weights to provide expert-guided output targets. These targets introduce expert-relative guidance into the quantization process, rather than passively reconstructing only the merged model. Together with this expert guidance, merged-weight anchoring stabilizes the calibration and preserves the integrated behavior of the merged model. The expert models are accessed only during the post-merge calibration stage. After quantization, the deployed model remains a single low-bit merged model, without experts or additional inference-time modules. Figure 1 illustrates this distinction.

We evaluate E-PMQ on CLIP, FLAN-T5, and Llama, spanning vision and language from 8-task CLIP-ViT-B/32 to 20-task CLIP-ViT-L/14 and models up to 8B parameters. Across these regimes, E-PMQ consistently outperforms naive merge-then-quantize PMQ. We further compare against expert-aware alternatives, including quantize-then-merge (HDRQ) and representation repair followed by quantization (Surgery + GPTQ). E-PMQ attains higher accuracy than these pipelines while remaining a single-stage calibration procedure that deploys one low-bit model without auxiliary inference-time modules.

We summarize the main contributions of this work as follows:

❶ We formulate *Post-Merge Quantization* (PMQ) as a distinct low-bit deployment setting for merged models, and identify a key failure mode of naive PMQ: directly reconstructing the merged model couples the *quantization deviation* introduced by low-bit reconstruction with the *expert-relative merging deviation* inherited from model merging.

❷ We introduce *E-PMQ*, an expert-guided PMQ framework that uses source expert weights to provide expert-guided output targets during layer-wise calibration, together with merged-weight anchoring to stabilize the calibration and preserve the integrated behavior of the merged model.

❸ We empirically validate E-PMQ on CLIP, FLAN-T5, and Llama (up to 8B). Relative to naive merge-then-quantize PMQ and to expert-aware alternatives including HDRQ and Surgery + GPTQ,

E-PMQ yields a more accurate deployed model via single-stage calibration, without auxiliary inference-time modules.

2 Related Work

Model Merging. Model merging composes multiple specialized models into a single model without joint training or deploying one model per task. Representative methods include weight averaging, Fisher merging, task arithmetic, TIES-Merging, DARE, and WUDI (Wortsman et al., 2022; Matena & Raffel, 2022; Ilharco et al., 2023; Yadav et al., 2023; Yu et al., 2024; Cheng et al., 2025). Methods based on feature statistics or feature drift, including RegMean, RegMean++, and LOT Merging, are closer to E-PMQ in mechanism: they derive layer updates during merging from feature statistics, regression objectives, or explicit feature-drift minimization (Jin et al., 2023; Nguyen et al., 2026; Sun et al., 2025). These methods define how to construct the merged model. In contrast, E-PMQ starts from an already merged model and uses source experts only during post-merge low-bit calibration.

Post-training quantization. PTQ compresses a trained full-precision model into low-bit weights without end-to-end retraining, typically by calibration-based rounding or layer-wise reconstruction (Nagel et al., 2020; Li et al., 2021; Frantar et al., 2023; Lin et al., 2024; Xiao et al., 2023; Yao et al., 2022). The standard assumption is that the full-precision model is a reliable target. That assumption is natural for independently trained models and is the reason naive PMQ applies GPTQ or AWQ directly to W_m. We argue that this target is often imperfect after merging.

Low-bit models from multiple experts. A single low-bit model can be obtained from multiple experts either before or after merging. Naive PMQ applies PTQ to a given merger W_m. HDRQ (Shin et al., 2025) quantizes experts before merging; TVQ (Kim et al., 2025) compresses task vectors and is complementary, since the merger can still be quantized post hoc. Surgery + GPTQ (Yang et al., 2024; Frantar et al., 2023) repairs representations and then quantizes the backbone, but keeps an extra module at inference. E-PMQ is post-merge, yet uses one-stage expert-guided calibration and serves one low-bit network. Native MoE quantization (Hu et al., 2025) retains routed experts and is a different deployed object. We compare these pipelines in Section 5.5 and discuss them further in App. B.

3 Preliminaries and Problem Formulation

Notation. Let $\{W_i\}_{i=1}^K$ denote K task-specialized expert models, and $W_m = \mathcal{M}(\{W_i\}_{i=1}^K)$ be the merged model produced by a merging algorithm $\mathcal{M}$. We use W_i^ℓ, W_m^ℓ, and Q^ℓ to denote the layer-ℓ weights of expert i, the merged model, and the quantized model, respectively. Let $\mathcal{D}_{\text{cal}}$ be a small calibration set, and let $X^\ell \in \mathbb{R}^{d_{\text{in}} \times n}$ denote the calibration activations entering layer ℓ, where n is the number of calibration tokens. The feasible set of b-bit quantized weights is denoted by $\mathcal{Q}_b$.

Post-Training Quantization. Post-training quantization compresses a full-precision model into a low-bit model using a small calibration set, without end-to-end retraining. For a generic full-precision model W, a PTQ algorithm produces a quantized model

$$Q = \mathcal{A}_{\text{ptq}}(W; \mathcal{D}_{\text{cal}}), \tag{1}$$

where $\mathcal{A}_{\text{ptq}}$ denotes the PTQ algorithm and Q is the resulting b-bit model.

Following the layer-wise reconstruction formulation used in GPTQ (Frantar et al., 2023), a reconstruction-based PTQ method minimizes the following layer-wise objective:

$$\min_{Q^\ell \in \mathcal{Q}_b} \left\| Q^\ell X^\ell - W^\ell X^\ell \right\|_F^2. \tag{2}$$

Accordingly, we characterize the layer-wise **quantization deviation** as

$$\Delta_{\text{quant}}^\ell(X^\ell) = Q^\ell X^\ell - W^\ell X^\ell. \tag{3}$$

Model Merging. Model merging combines multiple task- or domain-specialized experts into a single model without joint training or deploying one model per task:

$$W_m = \mathcal{M}\left(\{W_i\}_{i=1}^{K}\right). \tag{4}$$

Since W_m is obtained by parameter composition, its intermediate representations may deviate from those of the original experts. Prior work has observed such representation-level discrepancy between merged models and source experts during model merging (Yang et al., 2024). Following this view, we characterize the layer-wise **expert-relative merging deviation** in the output space. We use X^ℓ as a common layer-wise input, which isolates the output discrepancy induced by different layer weights under the same inputs. The deviation of the merged layer from expert i is

$$\Delta^\ell_{\mathrm{merge},i}(X^\ell) = W^\ell_m X^\ell - W^\ell_i X^\ell. \tag{5}$$

This term measures how far the merged model has moved away from the behavior of each source expert before quantization is applied.

Post-Merge Quantization. In this work, we formulate post-merge quantization, where the goal is to obtain a low-bit model after merging. PMQ produces a quantized merged model

$$Q_m = \mathcal{A}_{\mathrm{pmq}}(W_m, \{W_i\}_{i=1}^{K}; \mathcal{D}_{\mathrm{cal}}), \tag{6}$$

where $\mathcal{A}_{\mathrm{pmq}}$ denotes a post-merge quantization algorithm. A straightforward solution is to directly apply a standard PTQ algorithm to the merged model:

$$Q^{\mathrm{naive}}_m = \mathcal{A}_{\mathrm{ptq}}(W_m; \mathcal{D}_{\mathrm{cal}}). \tag{7}$$

At layer ℓ, naive PMQ follows a standard layer-wise output-reconstruction objective and minimizes

$$\min_{Q^\ell \in \mathcal{Q}_b} \left\| Q^\ell X^\ell - W^\ell_m X^\ell \right\|_F^2. \tag{8}$$

However, this objective treats the full-precision merged model as a reliable standalone reconstruction target. This assumption is problematic in PMQ because the merged model may already contain expert-relative merging deviations before quantization.

To make this deviation explicit, consider the output deviation of the quantized merged layer with respect to expert i:

$$Q^\ell X^\ell - W^\ell_i X^\ell = \underbrace{Q^\ell X^\ell - W^\ell_m X^\ell}_{\text{quantization deviation}} + \underbrace{W^\ell_m X^\ell - W^\ell_i X^\ell}_{\text{expert-relative merging deviation}}. \tag{9}$$

The first term is the reconstruction error that ordinary PTQ minimizes. The second is inherited from merging and is invisible to any objective that only fits W^ℓ_m. Write $D^\ell_{\mathrm{quant}} = Q^\ell X^\ell - W^\ell_m X^\ell$ and $D^\ell_{\mathrm{merge},i} = W^\ell_m X^\ell - W^\ell_i X^\ell$. Naive PMQ therefore leaves the merging deviation untouched while introducing a quantization term. What later layers actually see is the **sum** of the two terms, $Q^\ell X^\ell - W^\ell_i X^\ell$. A PMQ method should reduce this accumulated residual rather than treating W_m as the sole target. How far Q should move away from W_m depends on how large the inherited merging deviation is: a weak merger leaves more room to repair, while a strong merger should stay close to W_m. This is the motivation for using source experts during quantization, together with an anchor to the merged weights.

4 METHOD

4.1 OVERVIEW

E-PMQ quantizes a full-precision merged model in forward, layer-wise order, using the source experts only as calibration-time teachers. When layer ℓ is quantized, earlier layers are already quantized, so the activations are those of the current partially quantized merged model.

Section 3 uses a shared X^ℓ to isolate weight-induced discrepancy. In calibration we instead collect a per-expert activation X^ℓ_i by running the current merged trajectory on the calibration subset of expert i. The target $Y^\ell_i = W^\ell_i X^\ell_i$ therefore applies expert weights to merged-path activations. E-PMQ matches these targets while anchoring Q^ℓ to W^ℓ_m in weight space.

4.2 EXPERT-GUIDED OBJECTIVE

E-PMQ reconstructs layer outputs on the current merged-path activations, but replaces the merged target $W_m^\ell X_i^\ell$ with the expert target

$$Y_i^\ell = W_i^\ell X_i^\ell. \tag{10}$$

Matching these targets alone,

$$\min_{Q^\ell \in \mathcal{Q}_b} \sum_{i=1}^{K} \left\| Q^\ell X_i^\ell - W_i^\ell X_i^\ell \right\|_F^2, \tag{11}$$

can over-correct when experts disagree. We therefore add a merged-weight anchor, a weight-space regularizer rather than a second output-reconstruction term:

$$\min_{Q^\ell \in \mathcal{Q}_b} \sum_{i=1}^{K} \left\| Q^\ell X_i^\ell - W_i^\ell X_i^\ell \right\|_F^2 + \lambda^\ell \left\| Q^\ell - W_m^\ell \right\|_F^2. \tag{12}$$

The first term pulls Q^ℓ toward expert outputs on the merged-path activations. The second keeps Q^ℓ near W_m^ℓ, so the solution cannot collapse onto a single expert. Their relative weight is the merge-quality trade-off described above: more repair when W_m is a weak target, more anchoring when it is already strong.

4.3 ADAPTIVE MERGED-WEIGHT ANCHORING

The scalar λ^ℓ sets this trade-off. Because layers differ in activation scale, we couple the weight-space penalty to the calibration energy:

$$\lambda^\ell = \frac{\alpha}{d_\ell} \operatorname{Tr}\left(\sum_{i=1}^{K} X_i^\ell (X_i^\ell)^\top \right) = \frac{\alpha}{d_\ell} \sum_{i=1}^{K} \left\| X_i^\ell \right\|_F^2, \tag{13}$$

where d_ℓ is the input dimension of layer ℓ and α is a global scalar. Setting $\alpha = 0$ removes the anchor and, as we show later, collapses accuracy.

4.4 CURVATURE-AWARE DISCRETE OPTIMIZATION

Eq. (12) defines a discrete layer-wise quadratic problem. Its formulation is independent of a particular post-training quantization algorithm: E-PMQ specifies the expert-aware target statistics and merged-weight anchoring, while a curvature-aware discrete optimizer selects the final low-bit weights. To expose the resulting quadratic statistics, define

$$H_i^\ell = X_i^\ell (X_i^\ell)^\top, \qquad H_q^\ell = \sum_{i=1}^{K} H_i^\ell. \tag{14}$$

Under the E-PMQ objective, the corresponding effective curvature and right-hand side are

$$H_{\text{E-PMQ}}^\ell = H_q^\ell + \lambda^\ell I, \qquad R_{\text{E-PMQ}}^\ell = \sum_{i=1}^{K} W_i^\ell H_i^\ell + \lambda^\ell W_m^\ell. \tag{15}$$

Here, $H_{\text{E-PMQ}}^\ell$ is the empirical curvature of the layer-wise output-reconstruction objective, and $R_{\text{E-PMQ}}^\ell$ is its expert-aware linear term. The term $\sum_{i=1}^{K} W_i^\ell H_i^\ell$ is induced by the expert-guided output targets, while $\lambda^\ell I$ and $\lambda^\ell W_m^\ell$ arise from merged-weight anchoring. These statistics define a solver interface for curvature-aware discrete quantization. For all reported experiments, we instantiate this interface with a sequential second-order rounding procedure following GPTQ's implementation scheme. The full procedure is summarized in App. C; the continuous relaxation, stationary condition, and closed-form relaxed optimizer are in App. D.

5 EXPERIMENTS

We evaluate E-PMQ along three axes: effectiveness across mergers, model families, and scales; evidence for its two objective terms; and calibration and deployment cost.

5.1 SETUP

Benchmarks. We evaluate three settings. CLIP-ViT-B/32 and CLIP-ViT-L/14 (Radford et al., 2021) follow the FusionBench (Tang et al., 2025) image-classification suites: the standard 8-task mix (SUN397, Stanford Cars, RESISC45, EuroSAT, SVHN, GTSRB, MNIST, and DTD (Xiao et al., 2010; Krause et al., 2013; Cheng et al., 2017; Helber et al., 2019; Netzer et al., 2011; Stallkamp et al., 2011; LeCun et al., 1998; Cimpoi et al., 2014)) and the 14- and 20-task extensions. FLAN-T5-base (Raffel et al., 2020; Wei et al., 2022; Chung et al., 2024) is evaluated on eight GLUE tasks (Wang et al., 2019): CoLA, MNLI, MRPC, QNLI, QQP, RTE, SST-2, and STS-B. Llama-3.2-3B-Instruct and Llama-3.1-8B-Instruct are evaluated on GSM8K, MATH500, ARC-C, IFEval, HumanEval, and MBPP+.

Merging and quantization methods. On 8-task CLIP we use Simple Averaging, Task Arithmetic, TIES-Merging, and WUDI-Merging; the 14- and 20-task CLIP suites and FLAN-T5 use Task Arithmetic and TIES-Merging; Llama uses Task Arithmetic. Naive PMQ applies RTN, GPTQ (Frantar et al., 2023), and AWQ (Lin et al., 2024) directly to the merged model (GPTQ and AWQ only for Llama). We compare alternative pipelines, including HDRQ and Surgery + GPTQ, in Section 5.5.

Quantization protocol. Unless otherwise specified, all settings use 4-bit weight-only quantization. CLIP and FLAN-T5 use 256 calibration samples per task ($256K$ samples for a K-task merger); remaining FusionBench hyperparameters are in App. K. Llama uses 256 samples per expert, group size 32, and a maximum calibration sequence length of 512; further LLM settings are in App. F. E-PMQ quantizes layer-wise in forward order on the same calibration data as the PTQ baselines.

5.2 CLIP: EFFECTIVENESS AND SCALING

Table 1: Performance of 4-bit post-merge quantized CLIP-ViT-B/32 models on eight image-classification tasks. All numbers are top-1 accuracy (%). Gray arrows show changes over the corresponding full-precision merged checkpoints.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Pre-trained	63.2	59.8	60.7	46.0	31.6	32.5	48.2	43.9	48.2
Fine-tuned (STL)	75.0	78.3	95.2	99.0	97.3	98.9	99.6	79.7	90.3
Simple Averaging	65.4$_{\uparrow 0.0}$	62.6$_{\uparrow 0.0}$	70.8$_{\uparrow 0.0}$	76.9$_{\uparrow 0.0}$	64.5$_{\uparrow 0.0}$	54.9$_{\uparrow 0.0}$	86.3$_{\uparrow 0.0}$	50.9$_{\uparrow 0.0}$	66.5$_{\uparrow 0.0}$
w/ RTN	62.0$_{\downarrow 3.4}$	54.7$_{\downarrow 7.9}$	65.6$_{\downarrow 5.2}$	60.2$_{\downarrow 16.7}$	61.6$_{\downarrow 2.9}$	47.7$_{\downarrow 7.2}$	84.4$_{\downarrow 1.9}$	46.3$_{\downarrow 4.6}$	60.3$_{\downarrow 6.2}$
w/ GPTQ	64.3$_{\downarrow 1.2}$	59.5$_{\downarrow 3.1}$	69.3$_{\downarrow 1.5}$	68.0$_{\downarrow 8.9}$	64.0$_{\downarrow 0.5}$	49.1$_{\downarrow 5.8}$	86.4$_{\uparrow 0.1}$	49.4$_{\downarrow 1.5}$	63.7$_{\downarrow 2.8}$
w/ AWQ	61.7$_{\downarrow 3.7}$	51.7$_{\downarrow 10.9}$	64.5$_{\downarrow 6.3}$	57.8$_{\downarrow 19.1}$	63.8$_{\downarrow 0.7}$	49.6$_{\downarrow 5.3}$	84.3$_{\downarrow 2.0}$	46.7$_{\downarrow 4.2}$	60.0$_{\downarrow 6.5}$
w/ E-PMQ	67.7$_{\uparrow 2.3}$	67.8$_{\uparrow 5.2}$	80.5$_{\uparrow 9.7}$	68.6$_{\downarrow 8.3}$	94.9$_{\uparrow 30.4}$	59.5$_{\uparrow 4.6}$	99.0$_{\uparrow 12.7}$	63.1$_{\uparrow 12.2}$	**75.1**$_{\uparrow 8.6}$
Task Arithmetic ($\lambda = 0.3$)	57.1$_{\uparrow 0.0}$	55.7$_{\uparrow 0.0}$	64.9$_{\uparrow 0.0}$	76.7$_{\uparrow 0.0}$	77.9$_{\uparrow 0.0}$	68.5$_{\uparrow 0.0}$	96.1$_{\uparrow 0.0}$	47.2$_{\uparrow 0.0}$	68.0$_{\uparrow 0.0}$
w/ RTN	52.0$_{\downarrow 5.1}$	45.7$_{\downarrow 10.0}$	59.9$_{\downarrow 5.0}$	62.8$_{\downarrow 13.9}$	75.4$_{\downarrow 2.5}$	54.6$_{\downarrow 13.9}$	95.1$_{\downarrow 1.0}$	44.3$_{\downarrow 2.9}$	61.2$_{\downarrow 6.8}$
w/ GPTQ	55.6$_{\downarrow 1.5}$	53.5$_{\downarrow 2.2}$	63.8$_{\downarrow 1.1}$	69.1$_{\downarrow 7.6}$	77.9$_{\uparrow 0.0}$	57.0$_{\downarrow 11.5}$	95.9$_{\downarrow 0.2}$	47.5$_{\uparrow 0.3}$	65.0$_{\downarrow 3.0}$
w/ AWQ	52.2$_{\downarrow 4.9}$	46.3$_{\downarrow 9.4}$	60.3$_{\downarrow 4.6}$	58.6$_{\downarrow 18.1}$	74.9$_{\downarrow 3.1}$	55.6$_{\downarrow 12.9}$	95.3$_{\downarrow 0.8}$	43.7$_{\downarrow 3.5}$	60.9$_{\downarrow 7.1}$
w/ E-PMQ	67.0$_{\uparrow 9.9}$	64.4$_{\uparrow 8.7}$	78.5$_{\uparrow 13.6}$	66.3$_{\downarrow 10.4}$	94.8$_{\uparrow 16.9}$	57.0$_{\downarrow 11.5}$	98.9$_{\uparrow 2.8}$	62.0$_{\uparrow 14.8}$	**73.6**$_{\uparrow 5.6}$
TIES-Merging ($\lambda = 0.3$)	67.1$_{\uparrow 0.0}$	64.2$_{\uparrow 0.0}$	74.1$_{\uparrow 0.0}$	76.8$_{\uparrow 0.0}$	77.7$_{\uparrow 0.0}$	69.4$_{\uparrow 0.0}$	94.1$_{\uparrow 0.0}$	54.0$_{\uparrow 0.0}$	72.2$_{\uparrow 0.0}$
w/ RTN	63.1$_{\downarrow 4.0}$	55.0$_{\downarrow 9.2}$	69.3$_{\downarrow 4.8}$	58.9$_{\downarrow 17.9}$	75.2$_{\downarrow 2.5}$	59.8$_{\downarrow 9.6}$	93.1$_{\downarrow 1.0}$	48.7$_{\downarrow 5.3}$	65.4$_{\downarrow 6.8}$
w/ GPTQ	65.6$_{\downarrow 1.5}$	61.1$_{\downarrow 3.1}$	72.3$_{\downarrow 1.8}$	67.7$_{\downarrow 9.1}$	76.7$_{\downarrow 1.0}$	62.9$_{\downarrow 6.5}$	93.9$_{\downarrow 0.2}$	53.0$_{\downarrow 1.0}$	69.1$_{\downarrow 3.1}$
w/ AWQ	62.9$_{\downarrow 4.2}$	55.4$_{\downarrow 8.8}$	68.8$_{\downarrow 5.3}$	56.6$_{\downarrow 20.2}$	76.2$_{\downarrow 1.5}$	59.3$_{\downarrow 10.1}$	93.3$_{\downarrow 0.8}$	49.3$_{\downarrow 4.7}$	65.2$_{\downarrow 7.0}$
w/ E-PMQ	67.6$_{\uparrow 0.5}$	66.7$_{\uparrow 2.5}$	80.5$_{\uparrow 6.4}$	67.2$_{\downarrow 9.6}$	94.7$_{\uparrow 17.0}$	59.2$_{\downarrow 10.2}$	99.0$_{\uparrow 4.9}$	63.2$_{\uparrow 9.2}$	**74.8**$_{\uparrow 2.6}$
WUDI-Merging	68.0$_{\uparrow 0.0}$	72.5$_{\uparrow 0.0}$	85.0$_{\uparrow 0.0}$	94.6$_{\uparrow 0.0}$	94.8$_{\uparrow 0.0}$	94.9$_{\uparrow 0.0}$	99.3$_{\uparrow 0.0}$	66.6$_{\uparrow 0.0}$	**84.5**$_{\uparrow 0.0}$
w/ RTN	62.9$_{\downarrow 5.1}$	63.8$_{\downarrow 8.7}$	79.4$_{\downarrow 5.6}$	85.5$_{\downarrow 9.1}$	94.2$_{\downarrow 0.6}$	80.0$_{\downarrow 14.9}$	99.1$_{\downarrow 0.2}$	59.6$_{\downarrow 7.0}$	78.1$_{\downarrow 6.4}$
w/ GPTQ	66.6$_{\downarrow 1.4}$	68.9$_{\downarrow 3.6}$	83.8$_{\downarrow 1.2}$	90.0$_{\downarrow 4.6}$	94.8$_{\uparrow 0.0}$	81.3$_{\downarrow 13.6}$	99.3$_{\uparrow 0.0}$	64.2$_{\downarrow 2.4}$	81.1$_{\downarrow 3.4}$
w/ AWQ	62.5$_{\downarrow 5.5}$	63.8$_{\downarrow 8.8}$	79.7$_{\downarrow 5.3}$	83.9$_{\downarrow 10.7}$	93.9$_{\downarrow 0.9}$	80.2$_{\downarrow 14.7}$	99.1$_{\downarrow 0.2}$	59.7$_{\downarrow 6.9}$	77.8$_{\downarrow 6.7}$
w/ E-PMQ	67.9$_{\downarrow 0.1}$	68.9$_{\downarrow 3.6}$	86.8$_{\uparrow 1.8}$	92.9$_{\downarrow 1.7}$	95.3$_{\uparrow 0.5}$	80.6$_{\downarrow 14.4}$	99.3$_{\uparrow 0.0}$	68.0$_{\uparrow 1.4}$	82.4$_{\downarrow 2.1}$

Effect across upstream mergers. Table 1 first tests whether E-PMQ remains effective as upstream merge quality varies. It is the strongest 4-bit method for all four mergers. Relative to GPTQ, E-PMQ gains 11.4, 8.6, and 5.7 points for Simple Averaging, Task Arithmetic, and TIES-Merging and exceeds their full-precision checkpoints. For the stronger WUDI merger, it remains the best quantized method (82.4) but stays below full precision (84.5). Thus, larger gains coincide with weaker merged targets, consistent with the correction– preservation trade-off motivating expert guidance and anchoring.

Scaling across backbones and task counts. Table 2 then tests whether this advantage persists with a larger backbone and more tasks. E-PMQ leads on 8-task ViT-L/14 and on both backbones with 14 and 20 tasks. The largest separation occurs for 20-task Task Arithmetic: GPTQ obtains 35.0 and 34.8 on B/32 and L/14, whereas E-PMQ reaches 64.2 and 76.7. The benefit therefore persists across model scale and grows when more-task merging produces a weaker target. Per-task results are in Apps. E.2 and E.3; 8-task L/14 results are in App. E.1.

Table 2: Extended CLIP average accuracy under 4-bit PMQ (8-task L/14; 14- and 20-task B/32 and L/14).

Method	8 Task L/14	14 Task B/32	14 Task L/14	20 Task B/32	20 Task L/14
Pre-trained	64.6	58.8	69.1	55.6	65.6
Fine-tuned	94.3	90.0	92.8	90.3	93.1
Task Arithmetic	80.7↑0.0	52.8↑0.0	63.1↑0.0	36.3↑0.0	57.2↑0.0
w/ RTN	77.3↓3.4	47.6↓5.2	57.8↓5.3	33.1↓3.2	33.7↓23.5
w/ GPTQ	78.3↓2.4	49.9↓2.9	59.2↓3.9	35.0↓1.3	34.8↓22.4
w/ AWQ	66.9↓13.8	48.3↓4.5	49.1↓14.1	34.0↓2.3	27.7↓29.5
w/ E-PMQ	**85.9**↑5.2	**70.1**↑17.3	**82.0**↑18.9	**64.2**↑27.9	**76.7**↑19.5
TIES-Merging	84.0↑0.0	67.6↑0.0	77.8↑0.0	55.6↑0.0	63.0↑0.0
w/ RTN	81.2↓2.8	60.8↓6.8	72.7↓5.1	51.0↓4.6	60.3↓2.7
w/ GPTQ	81.8↓2.2	63.5↓4.1	74.0↓3.8	53.1↓2.5	61.1↓1.9
w/ AWQ	73.4↓10.6	60.8↓6.8	67.9↓9.9	51.4↓4.2	58.0↓5.0
w/ E-PMQ	**86.0**↑2.0	**72.1**↑4.5	**82.5**↑4.7	**67.8**↑12.2	**77.5**↑14.5

5.3 Language: Cross-Architecture Generalization

Table 3: FLAN-T5 results under 4-bit post-merge quantization on GLUE tasks. All numbers are task scores (%), and STS-B reports Spearman's correlation. Gray arrows show changes over the corresponding full-precision merged checkpoints.

Method	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Avg.
Task Arithmetic	69.80↑0.0	57.66↑0.0	78.43↑0.0	90.26↑0.0	83.61↑0.0	80.51↑0.0	92.20↑0.0	77.82↑0.0	78.79↑0.0
w/ RTN	70.18↑0.4	56.29↓1.4	78.92↑0.5	90.04↓0.2	83.78↑0.2	81.59↑1.1	91.17↓1.0	76.10↓1.7	78.51↓0.3
w/ GPTQ	69.42↓0.4	55.99↓1.7	77.70↓0.7	89.90↓0.4	83.73↑0.1	79.42↓1.1	92.20↑0.0	77.74↓0.1	78.26↓0.5
w/ AWQ	69.51↓0.3	61.30↑3.6	78.43↑0.0	89.84↓0.4	82.91↓0.7	80.51↑0.0	91.06↓1.1	76.85↓1.0	78.80↑0.0
w/ E-PMQ	69.80↑0.0	82.50↑24.8	79.41↑1.0	90.43↑0.2	84.34↑0.7	82.67↑2.2	92.78↑0.6	84.80↑7.0	**83.34**↑4.6
TIES-Merging	70.37↑0.0	65.02↑0.0	78.68↑0.0	90.24↑0.0	83.53↑0.0	81.59↑0.0	91.86↑0.0	78.58↑0.0	79.98↑0.0
w/ RTN	70.37↑0.0	63.57↓1.5	79.66↑1.0	89.91↓0.3	83.56↑0.0	82.67↑1.1	91.06↓0.8	75.42↓3.2	79.53↓0.5
w/ GPTQ	69.80↓0.6	65.87↑0.9	78.19↓0.5	89.95↓0.3	83.20↓0.3	80.51↓1.1	91.28↓0.6	78.45↓0.1	79.66↓0.3
w/ AWQ	68.55↓1.8	68.16↑3.1	78.43↓0.3	89.91↓0.3	82.81↓0.7	79.78↓1.8	90.83↓1.0	76.51↓2.1	79.37↓0.6
w/ E-PMQ	69.51↓0.9	82.48↑17.5	81.37↑2.7	90.81↑0.6	84.30↑0.8	80.51↓1.1	93.00↑1.1	85.84↑7.3	**83.48**↑3.5

We next test whether the CLIP result transfers to language models. On FLAN-T5-base (Table 3), E-PMQ leads under both Task Arithmetic and TIES-Merging (83.34 and 83.48) and exceeds the respective full-precision averages (78.79 and 79.98), with most of the Task Arithmetic gain coming from MNLI.

Table 4: Performance of 4-bit post-merge quantized Llama models merged by Task Arithmetic. All numbers are scores (%). Gray arrows show changes over the corresponding full-precision merged models.

Llama-3.2-3B-Instruct

Method	GSM8K	MATH500	ARC-C	IFEval	HumanEval	MBPP+	Avg.
Task Arithmetic	74.91↑0.0	43.20↑0.0	72.70↑0.0	62.48↑0.0	53.66↑0.0	57.94↑0.0	60.81↑0.0
w/ AWQ	74.45↓0.46	39.60↓3.60	72.44↓0.26	60.63↓1.85	51.22↓2.44	56.08↓1.86	59.07↓1.74
w/ GPTQ	73.77↓1.14	41.40↓1.80	70.90↓1.80	58.78↓3.70	51.83↓1.83	55.56↓2.38	58.71↓2.10
w/ E-PMQ	74.60↓0.31	44.60↑1.40	70.99↓1.71	62.48↑0.0	51.83↓1.83	57.14↓0.80	**60.27**↓0.54

Llama-3.1-8B-Instruct

Method	GSM8K	MATH500	ARC-C	IFEval	HumanEval	MBPP+	Avg.
Task Arithmetic	85.67↑0.0	48.20↑0.0	77.99↑0.0	50.09↑0.0	65.24↑0.0	61.38↑0.0	64.76↑0.0
w/ AWQ	84.31↓1.36	48.60↑0.40	76.62↓1.37	44.73↓5.36	58.54↓6.70	60.85↓0.53	62.27↓2.49
w/ GPTQ	85.14↓0.53	47.00↓1.20	76.45↓1.54	44.92↓5.17	58.54↓6.70	57.94↓3.44	61.66↓3.10
w/ E-PMQ	84.23↓1.44	45.80↓2.40	76.28↓1.71	48.80↓1.29	60.98↓4.26	61.38↑0.0	**62.91**↓1.85

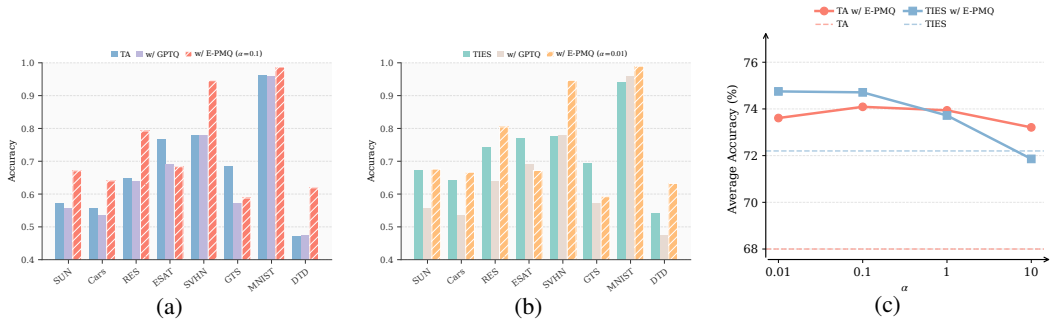


Figure 2: **Ablation analysis of E-PMQ.** (a) Task-level performance under Task Arithmetic (TA). (b) Task-level performance under TIES-Merging. (c) Sensitivity to the positive anchor strength α.

Table 5: Anchor ablation on CLIP-ViT-B/32.

Method	Avg.
Task Arithmetic	68.00↑0.0
w/ GPTQ	65.03↓3.0
w/o Anchor ($\alpha = 0$)	5.37↓62.6
w/ E-PMQ ($\alpha = 0.1$)	**74.09**↑6.1
TIES-Merging	72.20↑0.0
w/ GPTQ	69.13↓3.1
w/o Anchor ($\alpha = 0$)	4.57↓67.6
w/ E-PMQ ($\alpha = 0.01$)	**74.75**↑2.5

Table 6: Calibration cost of W4 E-PMQ on CLIP-ViT-B/32 (NVIDIA A800). All-at-once stores all experts, whereas Sequential loads them individually.

Experts	All-at-once		Sequential	
	Time (s)	GPU (GiB)	Time (s)	GPU (GiB)
2	27.70	3.73	36.34	2.33
4	33.34	5.72	48.62	2.37
6	40.08	7.71	62.49	2.38
8	53.84	9.69	76.65	2.38

On Llama (Table 4), E-PMQ remains the best quantized method at 3B and 8B (60.27 and 62.91), but stays below full precision (60.81 and 64.76). Together, these results transfer the E-PMQ ranking beyond vision while showing that recovery beyond full precision depends on the upstream merger.

5.4 Mechanism Analysis

We now examine the anchor and the joint residual targeted by E-PMQ.

Role of merged-weight anchoring. Figure 2 and Table 5 test the stabilizing role of merged-weight anchoring. Removing the anchor ($\alpha = 0$) collapses E-PMQ to 5.37 and 4.57 under Task Arithmetic and TIES-Merging, whereas positive α recovers 74.09 and 74.75 and remains stable over the evaluated range. This supports the anchor's role in preventing expert guidance from over-correcting the merger. Full per-task and bit-width results are in Apps. H and I.

Layer-wise joint residual. Figure 3 reports the layer-wise joint residual on merged-path calibration activations, $\frac{1}{K}\sum_i \|Q^\ell X_i^\ell - W_i^\ell X_i^\ell\|_F^2$. By Eq. (9), this metric captures the joint effect of quantization and expert-relative merging deviations. E-PMQ yields a smaller residual than GPTQ across layers, with larger gaps in later blocks, consistent with reducing the accumulated deviation seen downstream. This diagnostic does not isolate the two terms individually. A worse quantize-then-merge control in App. G further shows that expert access alone is insufficient.

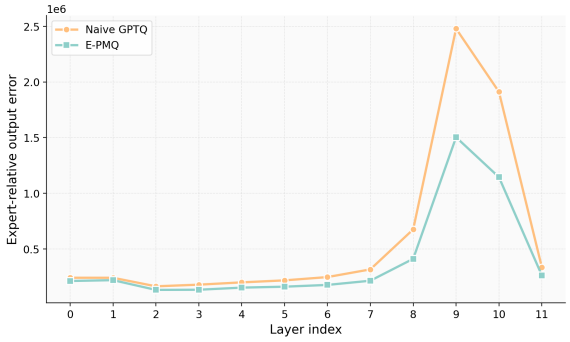


Figure 3: Layer-wise expert-relative residual on CLIP-ViT-B/32, averaged over experts on merged-path activations. Lower is better. By Eq. (9), this metric captures the joint effect of quantization and expert-relative merging deviations.

Table 7: Accuracy and expert-aware processing cost on eight-task CLIP-ViT-B/32 with Simple Averaging (W4A16). E-PMQ uses sequential loading; GPTQ is included as the naive PMQ accuracy reference. Extra denotes an auxiliary inference-time module.

Method	Pipeline	Avg.	Time (s)	Extra
HDRQ	Quantize-then-merge	56.54	8922.6	No
GPTQ	Merge-then-quantize	63.7	–	No
Surgery + GPTQ	Repair, then quantize	73.59	142.2	Yes
E-PMQ	One-stage PMQ	**75.10**	76.7	No

5.5 Efficiency and Deployment Trade-offs

We finally evaluate the calibration overhead of expert access and compare alternative routes to a W4 model.

Calibration memory and time. Table 6 shows that all-at-once peak GPU memory grows from 3.73 to 9.69 GiB for two to eight experts, while sequential loading stays near 2.38 GiB. At eight experts, it reduces peak allocation by 75.4% with a 42.4% calibration-time increase. This cost is calibration-only; deployment remains one W4 model independent of K.

Comparison with alternative pipelines. Table 7 compares four pipelines on eight-task CLIP-ViT-B/32. E-PMQ reaches 75.10 in 76.7s, outperforming Surgery + GPTQ (73.59; 142.2s), GPTQ (63.7), and HDRQ (56.54; 8922.6s). Unlike HDRQ and Surgery + GPTQ, E-PMQ uses one post-merge calibration stage and retains no auxiliary inference-time module. Its advantage over Surgery + GPTQ also holds on 3B and 8B Llama. Thus, E-PMQ incurs calibration access to experts while preserving the target deployment: one low-bit model. Full results and costs are in Apps. B and J.

Relation to full-precision repair. To address the joint effect of quantization deviation and expert-relative merging deviation, a natural alternative is to repair W_m in full precision and subsequently apply PTQ. This separates expert-aware calibration from low-bit reconstruction: unless expert statistics are reused, the PTQ stage again optimizes fidelity only to its full-precision input. E-PMQ instead retains expert targets and merged-weight anchoring within the low-bit layer objective, evaluated along the current partially quantized trajectory. Our claim is therefore not that PMQ replaces general merging repair, but that expert information should remain available when the final deployable low-bit weights are selected.

6 Conclusion

We formulate post-merge quantization (PMQ) as low-bit deployment after model merging and identify why naive PMQ is unreliable: reconstructing only the merged weights W_m leaves expert-relative merging deviation unaddressed while adding quantization deviation. E-PMQ mitigates their joint effect through one-stage layer-wise calibration, combining expert-guided output targets with adaptive merged-weight anchoring to avoid over-correcting the merger. Experts are used only for calibration; deployment remains a single low-bit model.

Across CLIP, FLAN-T5, and Llama up to 8B, E-PMQ consistently outperforms naive PMQ. On the evaluated CLIP and Llama settings, it also improves over quantize-then-merge and repair-then-quantize alternatives without auxiliary inference-time modules. Gains are largest for weaker mergers; on strong WUDI and Llama mergers, E-PMQ remains the best quantized method but stays below full precision. Ablations validate anchoring, residual analysis supports the mechanism, and sequential loading keeps GPU memory nearly constant across 2–8 experts at additional calibration and storage cost. E-PMQ therefore provides a practical path from multiple experts to one low-bit merged model.

AI USE STATEMENT

We used generative AI tools to assist with language editing, organization, LaTeX formatting, and the presentation and interpretation of existing experimental results. The tools were not used to generate experimental results. The authors reviewed and verified all AI-assisted content and take responsibility for the final paper.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 specify the problem, objective, and solver. Section 5 and the appendices document the evaluation protocol, hyperparameters, calibration data, per-task results, ablations, and compute costs needed to reproduce the reported experiments.

REFERENCES

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. Whoever started the interference should end it: Guiding data-free model merging via task vectors. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10121–10143. PMLR, 2025.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

Adam Coates, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. *International Conference on Artificial Intelligence and Statistics*, 2011.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre van Schaik. Emnist: Extending mnist to handwritten letters. In *International Joint Conference on Neural Networks*, 2017.

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

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. Task singular vectors: Reducing task interference in model merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18695–18705, 2025.

Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, Radu Ionescu, Marius Popescu, Cristian Grozea, James Bergstra, Jingjing Xie, Lukasz Romaszko, Bing Xu, Zhang Chuang, and Yoshua Bengio. Challenges in representation learning: A report on three machine learning contests. *Neural Networks*, 64:59–63, 2015.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Xing Hu, Zhixuan Chen, Dawei Yang, Zukang Xu, Chen Xu, Zhihang Yuan, Sifan Zhou, and Jiangyong Yu. Moequant: Enhancing quantization for mixture-of-experts large language models via expert-balanced sampling and affinity guidance, 2025. URL https://arxiv.org/abs/2505.03804.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023.

Xisen Jin, Xiang Ren, Daniel Preoţiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FCnohuR6AnM.

Youngeun Kim, Seunghwan Lee, Aecheon Jung, Bogon Ryu, and Sungeun Hong. Task vector quantization for memory-efficient model merging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision Workshops*, 2013.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

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*, 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 llm compression and acceleration. In *Proceedings of Machine Learning and Systems*, 2024.

Michael S. Matena and Colin A. Raffel. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*, 2022.

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*, 2020.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

The-Hai Nguyen, Huu-Tien Dang, Takeshi Suzuki, and Le-Minh Nguyen. RegMean++: Enhancing effectiveness and generalization of regression mean for model merging. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=H5lDsSCS9i.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Juncheol Shin, Minsang Seok, Seonggon Kim, and Eunhyeok Park. Merge-friendly post-training quantization for multi-target domain adaptation. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing*, 2013.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: A multi-class classification competition. In *International Joint Conference on Neural Networks*, 2011.

Wenju Sun, Qingyong Li, Wen Wang, Yang Liu, Yangliao Geng, and Boyang Li. Towards minimizing feature drift in model merging: Layer-wise task vector fusion for adaptive knowledge integration. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5745.

Anke Tang, Li Shen, Yong Luo, Enneng Yang, Han Hu, Lefei Zhang, Bo Du, and Dacheng Tao. Fusionbench: A unified library and comprehensive benchmark for deep model fusion. *Journal of Machine Learning Research*, 26(307):1–38, 2025.

Bastiaan S. Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *Medical Image Computing and Computer Assisted Intervention*, 2018.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.

Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jiménez, François Fleuret, and Pascal Frossard. Localizing task information for improved model merging and compression. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. *International Conference on Learning Representations*, 2022.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, 2022.

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*, 2023.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, 2023.

Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. Representation surgery for multi-task model merging. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

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. In *Advances in Neural Information Processing Systems*, 2022.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch, 2024. URL https://arxiv.org/abs/2311.03099.

A LIMITATIONS

E-PMQ has several limitations. First, it requires access to the source experts during the pre-deployment quantization stage. If only the merged model is available, E-PMQ cannot construct expert-guided targets and reduces to direct post-merge quantization. This requirement also introduces additional quantization-stage memory and compute compared with naive PMQ baselines, since expert weights are used to form layer-wise output targets.

The cost of E-PMQ scales with the number of experts K and the calibration budget. App. J shows that sequential expert loading reduces this GPU-memory cost to approximately $O(1)$. This makes the method most suitable for scenarios where experts are available during compression and additional pre-deployment computation is acceptable. Importantly, this extra cost is not incurred at inference time: after quantization, E-PMQ produces a single low-bit merged model and does not require experts, calibration data, or additional inference-time modules.

E-PMQ also depends on the representativeness of the calibration set. If the calibration samples poorly cover the task distributions associated with the experts, the expert-guided targets may provide a weaker calibration signal. Extending the method to settings where source experts are unavailable, or to merging recipes beyond the ones studied here, remains future work.

B COMPRESSION-ORIENTED MERGING AND EXPERT-AWARE QUANTIZATION

This appendix expands the related-work discussion in the main text and reports the corresponding comparisons.

Positioning. Existing expert-aware quantization methods include TVQ, HDRQ, and MoEQuant, but they address different settings. TVQ (Kim et al., 2025) quantizes task vectors before merging and is therefore complementary to E-PMQ, which quantizes the final merged model. MoEQuant (Hu et al., 2025) targets native sparse Mixture-of-Experts models and uses expert-balanced sampling and token–expert affinity to correct calibration imbalance among routed experts. E-PMQ instead targets a single model merged from task-specific experts: it uses source experts only during calibration and ultimately deploys one low-bit model without expert branches or routing. HDRQ (Shin et al., 2025) quantizes each expert into a merge-friendly state before merging, reducing the storage cost of maintaining multiple experts prior to model merging. In contrast, PMQ starts from a given merged model and seeks to calibrate it into a high-performing low-bit model. HDRQ and E-PMQ therefore differ in both problem setting and objective.

Compression-oriented merging methods such as TALL Mask (Wang et al., 2024) and TSV (Gargiulo et al., 2025) act on the merge stage or on pre-merge task vectors. TALL Mask identifies task-relevant parameter regions and retains task-specific updates only in these regions. TSV represents task vectors using their dominant singular directions to compress task-specific updates and reduce directional interference. Both are complementary to E-PMQ: they change how the merged model is formed, while E-PMQ quantizes an already merged model. Ordinary PTQ such as GPTQ (Frantar et al., 2023) reconstructs a single full-precision model; E-PMQ reconstructs source-expert outputs while anchoring the quantized weights to the merged model. Table 8 summarizes this distinction.

Table 8: GPTQ reconstructs a single full-precision model, whereas E-PMQ reconstructs source-expert behavior while anchoring the quantized weights to the merged model.

Method	Target setting	Reconstruction objective	Error addressed
GPTQ	PTQ of a single pre-trained model	Preserve the full-precision model	Quantization deviation
E-PMQ	PMQ of an already merged model	Recover source-expert behavior and preserve the merged model	Merging and quantization deviation

Comparison with HDRQ. Despite the difference in setting, we reproduce HDRQ under the CLIP-ViT-B/32 eight-task protocol. All methods use the same eight source-expert checkpoints, test sets,

evaluation pipeline, W4A16 deployment precision, and Simple Averaging; method-specific calibration follows the corresponding official configuration. Table 9 shows that E-PMQ outperforms HDRQ on all eight tasks (75.10% vs. 56.54%).

Table 9: Comparison with HDRQ on CLIP-ViT-B/32 eight-task Simple Averaging under W4A16.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Simple Averaging	65.40	62.60	70.80	76.90	64.50	54.90	86.30	50.90	66.50
HDRQ	58.90	50.43	61.44	54.96	54.13	44.45	81.34	46.65	56.54
E-PMQ	67.70	67.80	80.50	68.60	94.90	59.50	99.00	63.10	**75.10**

We further analyze this gap from empirical and theoretical perspectives. Empirically, we evaluate each HDRQ-quantized expert on its corresponding task before merging (Table 10). The standalone experts lose only 1.76 points on average, and five of eight lose at most 1.05 points. Yet merging them yields 56.54% average accuracy, 9.96 points below full-precision Simple Averaging (66.50%). The main degradation therefore emerges after merging, rather than from severe deterioration of individual experts.

Table 10: Per-expert accuracy before merging under HDRQ. The standalone experts lose only 1.76 points on average, so the large drop after merging is not explained by severe expert-wise deterioration.

	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Full-precision expert	74.86	78.51	95.11	98.67	97.27	90.35	99.58	79.68	89.25
HDRQ expert	72.41	73.56	94.06	98.37	97.06	89.60	99.47	75.43	87.49
Change	−2.45	−4.95	−1.05	−0.30	−0.21	−0.75	−0.11	−4.26	−1.76

Theoretically, HDRQ does not explicitly align the final merged model with all source experts. Let $\widetilde{\theta}_j = \theta_j + \epsilon_j$ denote quantized expert j. HDRQ indirectly improves merge compatibility by reducing the quantized error barrier B_Q in Eq. (16) between two experts,

$$B_Q = \max_{\beta \in [0,1]} \Big[\mathcal{L}\big((1-\beta)\widetilde{\theta}_1 + \beta\widetilde{\theta}_2\big) - (1-\beta)\mathcal{L}(\widetilde{\theta}_1) - \beta\mathcal{L}(\widetilde{\theta}_2) \Big], \tag{16}$$

but this regularizes an upper bound on pairwise mergeability rather than the deployed merged model. E-PMQ instead performs the alignment directly at each layer, as in Eq. (12): the first term reconstructs expert outputs, and the second anchors the solution to the full-precision merged weight.

For quantization deviation under linear averaging, write $W_m = \sum_i \alpha_i W_i$ with $\sum_i \alpha_i = 1$, and let $E_i = Q_i(W_i) - W_i$. This expansion is for averaging-style mergers such as Simple Averaging; TIES and WUDI are not of this form. Quantize-then-merge produces the merged weight in Eq. (17),

$$\widehat{W}_{\mathrm{QTM}} = \sum_i \alpha_i Q_i(W_i) = W_m + \sum_i \alpha_i E_i, \tag{17}$$

whose reconstruction error on input X is given by Eq. (18):

$$D_{\mathrm{QTM}} = \Big\| \sum_i \alpha_i E_i X \Big\|_F^2 = \sum_i \alpha_i^2 \|E_i X\|_F^2 + \sum_{i \neq j} \alpha_i \alpha_j \langle E_i X,\, E_j X \rangle_F. \tag{18}$$

Independent expert-wise calibration does not minimize this aggregate error or control its cross-expert terms; correlated errors or signed merging coefficients can therefore amplify the final distortion. Naive merge-then-quantize (e.g., GPTQ on W_m) instead optimizes the deployed model itself, as in Eq. (19):

$$D^*_{\mathrm{MTQ}} = \min_{Q \in \mathcal{Q}_b} \|(Q - W_m)X\|_F^2. \tag{19}$$

If $\widehat{W}_{\mathrm{QTM}} \in \mathcal{Q}_b$, optimality gives $D^*_{\mathrm{MTQ}} \leq D_{\mathrm{QTM}}$ from Eqs. (18) and (19); otherwise, projection or requantization introduces additional distortion. Although quantize-then-merge is not necessarily worse in every special case, it does not directly minimize the aggregate error of the final deployed model. The bound $D^*_{\mathrm{MTQ}} \leq D_{\mathrm{QTM}}$ therefore compares naive PMQ with QTM, not E-PMQ with QTM. E-PMQ does not minimize D^*_{MTQ}; it replaces the merged-output target with expert-guided targets plus a merged-weight anchor.

Comparison with Surgery + GPTQ. Because E-PMQ jointly calibrates expert-relative merging deviation and quantization deviation in the PMQ setting, a strong baseline should address the same two deviations separately. Representation Surgery (Yang et al., 2024) followed by GPTQ provides such a baseline. Surgery freezes the merged backbone and trains lightweight task-specific modules to map merged representations toward the corresponding expert representations, thereby reducing representation bias after merging. GPTQ then quantizes the frozen merged backbone, while the learned Surgery modules are retained for representation correction. This staged pipeline closely matches the PMQ objective—first correcting merging deviation and then reducing quantization deviation—but handles the two deviations separately. Table 11 reports CLIP-ViT-B/32 eight-task W4 results under Simple Averaging: E-PMQ improves over Surgery + GPTQ by 1.51 points (75.10% vs. 73.59%). Unlike this two-stage baseline, E-PMQ reduces the accumulated residual within a single layer-wise calibration process and requires no separate representation-repair stage. Table 12 shows the same comparison on Llama models merged by Task Arithmetic. E-PMQ remains the best quantized method at both scales, improving over Surgery + GPTQ by 1.13 points on the 3B model and 0.62 points on the 8B model.

Table 11: Comparison with Surgery + GPTQ on CLIP-ViT-B/32 eight-task Simple Averaging under W4.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Surgery + GPTQ	66.05	58.62	78.76	92.00	68.32	66.60	96.68	61.65	73.59
E-PMQ	67.70	67.80	80.50	68.60	94.90	59.50	99.00	63.10	**75.10**

Table 12: Comparison with Surgery + GPTQ on Llama models merged by Task Arithmetic under 4-bit PMQ.

Model	Method	GSM8K	MATH500	ARC-C	IFEval	HumanEval	MBPP+	Avg.
Llama-3.2-3B-Instruct	Task Arithmetic (FP)	74.91	43.20	72.70	62.48	53.66	57.94	60.81
	w/ AWQ	74.45	39.60	72.44	60.63	51.22	56.08	59.07
	w/ GPTQ	73.77	41.40	70.90	58.78	51.83	55.56	58.71
	w/ Surgery + GPTQ	73.39	40.20	72.10	61.20	51.22	56.72	59.14
	w/ E-PMQ	74.60	44.60	70.99	62.48	51.83	57.14	**60.27**
Llama-3.1-8B-Instruct	Task Arithmetic (FP)	85.67	48.20	77.99	50.09	65.24	61.38	64.76
	w/ AWQ	84.31	48.60	76.62	44.73	58.54	60.85	62.27
	w/ GPTQ	85.14	47.00	76.45	44.92	58.54	57.94	61.66
	w/ Surgery + GPTQ	84.69	46.40	76.37	46.86	59.76	59.66	62.29
	w/ E-PMQ	84.23	45.80	76.28	48.80	60.98	61.38	**62.91**

Compatibility with TSV and TVQ. The E-PMQ objective in Eq. (12) does not assume linear averaging or any particular merging operator. The upstream method affects only the merged-weight anchor W_m^ℓ, while the expert-guided term and low-bit constraint remain unchanged. Table 13 evaluates E-PMQ after TSV and after a TVQ→Task Arithmetic pipeline. E-PMQ is the best quantized method in both cases, and it is strongest under TVQ→TA.

Table 13: E-PMQ after compression-oriented upstream merging on CLIP-ViT-B/32 eight-task classification.

Upstream pipeline	Candidate	Avg. Acc. (%)
TVQ → Task Arithmetic	FP	66.17
TVQ → Task Arithmetic	GPTQ	64.72
TVQ → Task Arithmetic	Surgery + GPTQ	72.96
TVQ → Task Arithmetic	E-PMQ	**73.56**
TSV	FP	81.18
TSV	GPTQ	80.07
TSV	Surgery + GPTQ	80.94
TSV	E-PMQ	**81.07**

C ALGORITHM

Algorithm 1 E-PMQ for Post-Merge Quantization

Require: Experts $\{W_i\}_{i=1}^K$, merging operator $\mathcal{M}$, calibration subsets $\{\mathcal{D}_{\rfloor\dashv\updownarrow,i}\}_{i=1}^K$, bit-width b, anchor scale α.

1: Obtain $W_m = \mathcal{M}(\{W_i\}_{i=1}^K)$ and initialize the current model from W_m.
2: **for** each layer $\ell = 1, \dots, L$ **do**
3: **for** each expert/task $i = 1, \dots, K$ **do**
4: Collect X_i^ℓ by running the current partially quantized merged model on $\mathcal{D}_{\rfloor\dashv\updownarrow,i}$, and set $H_i^\ell = X_i^\ell (X_i^\ell)^\top$.
5: **end for**
6: Set $H_q^\ell = \sum_{i=1}^K H_i^\ell$ and $\lambda^\ell = \frac{\alpha}{d_\ell} \sum_{i=1}^K \|X_i^\ell\|_F^2$.
7: Form $H_{\text{E-PMQ}}^\ell = H_q^\ell + \lambda^\ell I$ and $R_{\text{E-PMQ}}^\ell = \sum_{i=1}^K W_i^\ell H_i^\ell + \lambda^\ell W_m^\ell$.
8: Use a curvature-aware sequential rounding procedure (GPTQ-style in our implementation) to obtain $Q^\ell \in \mathcal{Q}_b$ under the E-PMQ objective.
9: Replace layer ℓ in the current model with Q^ℓ.
10: **end for**
11: **return** Quantized merged model Q_m.

D CONTINUOUS RELAXATION AND CURVATURE STATISTICS

We provide the continuous relaxation of the E-PMQ objective and derive the quadratic statistics used by the curvature-aware discrete optimizer. Although the deployed weight is obtained by sequential rounding in the discrete low-bit space, the continuous relaxation clarifies how expert-guided output targets and merged-weight anchoring modify the layer-wise quadratic problem.

Recall the E-PMQ objective for layer ℓ:

$$\min_{Q^\ell \in \mathcal{Q}_b} \sum_{i=1}^K \left\| Q^\ell X_i^\ell - W_i^\ell X_i^\ell \right\|_F^2 + \lambda^\ell \left\| Q^\ell - W_m^\ell \right\|_F^2 . \tag{20}$$

To obtain a continuous relaxation, we ignore the discrete constraint $Q^\ell \in \mathcal{Q}_b$. Define

$$H_i^\ell = X_i^\ell (X_i^\ell)^\top, \qquad H_q^\ell = \sum_{i=1}^K H_i^\ell. \tag{21}$$

Then the relaxed objective can be written as the following quadratic form:

$$\sum_{i=1}^K \mathrm{Tr}\left[(Q^\ell - W_i^\ell) H_i^\ell (Q^\ell - W_i^\ell)^\top \right] + \lambda^\ell \left\| Q^\ell - W_m^\ell \right\|_F^2 . \tag{22}$$

Taking the derivative with respect to Q^ℓ and setting it to zero gives the stationary condition

$$Q^\ell \left(H_q^\ell + \lambda^\ell I \right) = \sum_{i=1}^K W_i^\ell H_i^\ell + \lambda^\ell W_m^\ell. \tag{23}$$

Therefore, the continuous relaxed optimizer is

$$Q_{\text{cont}}^\ell = \left(\sum_{i=1}^K W_i^\ell H_i^\ell + \lambda^\ell W_m^\ell \right) \left(H_q^\ell + \lambda^\ell I \right)^{-1} . \tag{24}$$

When $\lambda^\ell > 0$, the anchoring term adds diagonal loading, making $H_q^\ell + \lambda^\ell I$ positive definite and the inverse well defined.

The continuous solution in Eq. (24) is not the deployed quantized weight. In practice, E-PMQ uses the corresponding quadratic form inside a curvature-aware sequential rounding procedure. The effective curvature used by the solver is

$$H_{\text{E-PMQ}}^\ell = H_q^\ell + \lambda^\ell I, \tag{25}$$

and the expert-guided right-hand side is

$$R^{\ell}_{\text{E-PMQ}} = \sum_{i=1}^{K} W_i^{\ell} H_i^{\ell} + \lambda^{\ell} W_m^{\ell}. \tag{26}$$

Here, the term $\sum_{i=1}^{K} W_i^{\ell} H_i^{\ell}$ is induced by expert-guided output targets, while $\lambda^{\ell} I$ and $\lambda^{\ell} W_m^{\ell}$ arise from merged-weight anchoring. These statistics are then used by the curvature-aware sequential rounding procedure described in Algorithm 1, instantiated with GPTQ-style rounding for all reported experiments.

E Additional CLIP Results

In addition to the 8-task CLIP-ViT-B/32 results in the main text, we report full per-task results for the larger backbone and extended task suites. The extended 14-task and 20-task settings include additional datasets such as Flowers102, PCAM, FER2013, Oxford-IIIT Pet, STL10, CIFAR100, CIFAR10, Food101, Fashion-MNIST, EMNIST Letters, KMNIST, and Rendered SST2 (Nilsback & Zisserman, 2008; Veeling et al., 2018; Goodfellow et al., 2015; Parkhi et al., 2012; Coates et al., 2011; Krizhevsky, 2009; Bossard et al., 2014; Xiao et al., 2017; Cohen et al., 2017; Clanuwat et al., 2018; Socher et al., 2013; Radford et al., 2021).

E.1 Full 8-Task Results on CLIP-ViT-L/14

Table 14: Performance of 4-bit post-merge quantized CLIP-ViT-L/14 models on eight image-classification tasks. All numbers are top-1 accuracy (%). Gray arrows show changes over the corresponding full-precision merged checkpoints.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Simple Averaging	72.5$_{\uparrow 0.0}$	81.5$_{\uparrow 0.0}$	82.2$_{\uparrow 0.0}$	90.0$_{\uparrow 0.0}$	81.6$_{\uparrow 0.0}$	74.0$_{\uparrow 0.0}$	96.6$_{\uparrow 0.0}$	61.8$_{\uparrow 0.0}$	80.0$_{\uparrow 0.0}$
w/ RTN	71.6$_{\downarrow 0.9}$	80.4$_{\downarrow 1.1}$	81.2$_{\downarrow 1.0}$	83.2$_{\downarrow 6.8}$	75.6$_{\downarrow 6.0}$	68.8$_{\downarrow 5.2}$	96.6$_{\uparrow 0.0}$	61.5$_{\downarrow 0.3}$	77.4$_{\downarrow 2.6}$
w/ GPTQ	72.0$_{\downarrow 0.5}$	80.5$_{\downarrow 1.0}$	81.8$_{\downarrow 0.4}$	86.5$_{\downarrow 3.5}$	80.8$_{\downarrow 0.8}$	69.0$_{\downarrow 5.0}$	96.7$_{\uparrow 0.1}$	61.7$_{\downarrow 0.1}$	78.6$_{\downarrow 1.4}$
w/ AWQ	66.6$_{\downarrow 5.9}$	63.4$_{\downarrow 18.1}$	68.5$_{\downarrow 13.7}$	66.0$_{\downarrow 24.0}$	58.9$_{\downarrow 22.7}$	52.3$_{\downarrow 21.7}$	89.8$_{\downarrow 6.8}$	55.0$_{\downarrow 6.8}$	65.1$_{\downarrow 14.9}$
w/ E-PMQ	76.2$_{\uparrow 3.7}$	88.3$_{\uparrow 6.8}$	89.0$_{\uparrow 6.8}$	87.4$_{\downarrow 2.6}$	97.1$_{\uparrow 15.5}$	76.8$_{\uparrow 2.8}$	99.2$_{\uparrow 2.6}$	74.2$_{\uparrow 12.4}$	**86.0**$_{\uparrow 6.0}$
Task Arithmetic	72.0$_{\uparrow 0.0}$	79.0$_{\uparrow 0.0}$	80.5$_{\uparrow 0.0}$	86.0$_{\uparrow 0.0}$	87.5$_{\uparrow 0.0}$	83.5$_{\uparrow 0.0}$	98.0$_{\uparrow 0.0}$	58.8$_{\uparrow 0.0}$	80.7$_{\uparrow 0.0}$
w/ RTN	71.0$_{\downarrow 1.0}$	77.0$_{\downarrow 2.0}$	79.6$_{\downarrow 0.9}$	77.1$_{\downarrow 8.9}$	84.6$_{\downarrow 3.0}$	72.7$_{\downarrow 10.8}$	97.9$_{\downarrow 0.1}$	58.8$_{\uparrow 0.0}$	77.3$_{\downarrow 3.4}$
w/ GPTQ	71.3$_{\downarrow 0.8}$	77.2$_{\downarrow 1.8}$	80.1$_{\downarrow 0.4}$	79.7$_{\downarrow 6.3}$	87.2$_{\downarrow 0.3}$	73.2$_{\downarrow 10.3}$	98.1$_{\uparrow 0.1}$	59.5$_{\uparrow 0.7}$	78.3$_{\downarrow 2.4}$
w/ AWQ	62.6$_{\downarrow 9.4}$	57.2$_{\downarrow 21.8}$	66.9$_{\downarrow 13.6}$	67.6$_{\downarrow 18.4}$	72.8$_{\downarrow 14.7}$	58.8$_{\downarrow 24.7}$	95.9$_{\downarrow 2.1}$	53.7$_{\downarrow 5.1}$	66.9$_{\downarrow 13.8}$
w/ E-PMQ	75.9$_{\uparrow 3.9}$	87.9$_{\uparrow 8.9}$	89.4$_{\uparrow 8.9}$	87.3$_{\uparrow 1.3}$	97.0$_{\uparrow 9.5}$	76.5$_{\downarrow 7.0}$	99.1$_{\uparrow 1.1}$	74.4$_{\uparrow 15.6}$	**85.9**$_{\uparrow 5.2}$
TIES-Merging	74.7$_{\uparrow 0.0}$	83.3$_{\uparrow 0.0}$	86.4$_{\uparrow 0.0}$	91.3$_{\uparrow 0.0}$	89.7$_{\uparrow 0.0}$	85.2$_{\uparrow 0.0}$	97.8$_{\uparrow 0.0}$	63.9$_{\uparrow 0.0}$	84.0$_{\uparrow 0.0}$
w/ RTN	73.8$_{\downarrow 0.9}$	81.5$_{\downarrow 1.8}$	85.4$_{\downarrow 1.0}$	85.2$_{\downarrow 6.1}$	86.6$_{\downarrow 3.1}$	76.3$_{\downarrow 8.9}$	97.8$_{\uparrow 0.0}$	62.7$_{\downarrow 1.2}$	81.2$_{\downarrow 2.8}$
w/ GPTQ	74.0$_{\downarrow 0.7}$	82.0$_{\downarrow 1.3}$	85.9$_{\downarrow 0.5}$	85.6$_{\downarrow 5.7}$	89.5$_{\downarrow 0.2}$	75.9$_{\downarrow 9.3}$	97.6$_{\downarrow 0.2}$	64.0$_{\uparrow 0.1}$	81.8$_{\downarrow 2.2}$
w/ AWQ	70.4$_{\downarrow 4.4}$	69.2$_{\downarrow 14.2}$	77.3$_{\downarrow 9.1}$	75.0$_{\downarrow 16.3}$	77.2$_{\downarrow 12.5}$	63.5$_{\downarrow 21.7}$	95.4$_{\downarrow 2.4}$	59.4$_{\downarrow 4.5}$	73.4$_{\downarrow 10.6}$
w/ E-PMQ	76.0$_{\uparrow 1.3}$	87.9$_{\uparrow 4.6}$	89.1$_{\uparrow 2.7}$	87.8$_{\downarrow 3.5}$	97.0$_{\uparrow 7.3}$	76.4$_{\downarrow 8.8}$	99.1$_{\uparrow 1.3}$	74.8$_{\uparrow 10.9}$	**86.0**$_{\uparrow 2.0}$
WUDI-Merging	80.3$_{\uparrow 0.0}$	90.8$_{\uparrow 0.0}$	94.1$_{\uparrow 0.0}$	98.4$_{\uparrow 0.0}$	97.0$_{\uparrow 0.0}$	98.0$_{\uparrow 0.0}$	99.3$_{\uparrow 0.0}$	80.4$_{\uparrow 0.0}$	**92.3**$_{\uparrow 0.0}$
w/ RTN	78.9$_{\downarrow 1.4}$	90.2$_{\downarrow 0.7}$	93.6$_{\downarrow 0.5}$	97.5$_{\downarrow 0.9}$	96.6$_{\downarrow 0.4}$	88.2$_{\downarrow 9.8}$	99.3$_{\uparrow 0.0}$	77.8$_{\downarrow 2.6}$	90.3$_{\downarrow 2.0}$
w/ GPTQ	79.1$_{\downarrow 1.2}$	90.2$_{\downarrow 0.6}$	94.0$_{\downarrow 0.2}$	97.7$_{\downarrow 0.7}$	96.9$_{\downarrow 0.1}$	88.4$_{\downarrow 9.6}$	99.3$_{\uparrow 0.0}$	78.5$_{\downarrow 1.9}$	90.5$_{\downarrow 1.8}$
w/ AWQ	74.1$_{\downarrow 6.2}$	78.5$_{\downarrow 12.3}$	85.4$_{\downarrow 8.7}$	92.6$_{\downarrow 5.8}$	94.9$_{\downarrow 2.1}$	84.3$_{\downarrow 13.7}$	98.4$_{\downarrow 0.9}$	71.6$_{\downarrow 8.8}$	85.0$_{\downarrow 7.3}$
w/ E-PMQ	79.9$_{\downarrow 0.4}$	91.0$_{\uparrow 0.2}$	95.1$_{\uparrow 1.0}$	97.82$_{\downarrow 0.6}$	97.3$_{\uparrow 0.3}$	88.9$_{\downarrow 9.1}$	99.3$_{\uparrow 0.0}$	80.8$_{\uparrow 0.4}$	91.3$_{\downarrow 1.0}$

Table 14 reports the full 8-task CLIP-ViT-L/14 results under 4-bit PMQ. E-PMQ consistently improves over naive PMQ baselines for Simple Averaging, Task Arithmetic, and TIES-Merging, and remains competitive for the stronger WUDI merged model.

E.2 Full 14-Task Results

Table 15 reports the full 14-task CLIP-ViT-B/32 results. E-PMQ substantially improves over RTN, GPTQ, and AWQ, especially for weaker upstream mergers such as Task Arithmetic.

Table 16 reports the full 14-task CLIP-ViT-L/14 results. The gains remain consistent on the larger CLIP backbone, showing that the improvement is not specific to CLIP-ViT-B/32.

Table 15: Full 14-task CLIP-ViT-B/32 results under 4-bit PMQ. All numbers are top-1 accuracy (%). Gray arrows in Avg. show changes over the corresponding full-precision merged checkpoints.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Flowers	PCAM	FER	Pets	STL10	C100	Avg.
Simple Averaging	64.80	60.40	67.10	67.00	50.70	45.60	76.60	46.90	67.40	65.20	51.60	84.20	97.20	70.40	65.40↑0.0
w/ RTN	61.63	53.75	62.06	53.07	50.48	42.64	73.41	43.51	60.56	52.38	39.91	86.40	96.45	64.97	60.09↓5.3
w/ GPTQ	63.89	58.34	66.13	60.04	49.58	44.17	75.09	47.61	64.35	64.39	38.02	88.01	96.63	69.19	63.24↓2.2
w/ AWQ	61.32	51.00	60.59	49.22	51.15	43.95	73.86	46.22	59.88	53.30	41.21	86.40	96.05	64.40	59.90↓5.5
w/ E-PMQ	65.65	63.95	75.63	64.00	91.46	54.32	97.89	57.39	74.03	75.44	39.51	90.30	95.10	74.09	**72.77**↑7.4
Task Arithmetic	41.80	33.20	47.30	55.40	46.50	48.40	88.70	37.00	38.60	64.10	46.10	65.90	84.60	41.70	52.80↑0.0
w/ RTN	37.93	24.90	41.73	47.96	45.45	41.37	86.23	34.63	32.38	54.20	31.32	66.88	82.76	38.82	47.61↓5.2
w/ GPTQ	40.34	31.05	45.76	51.44	46.44	44.19	87.43	36.70	35.92	58.14	28.71	68.60	83.58	40.50	49.92↓2.9
w/ AWQ	37.97	27.16	42.56	47.04	44.58	40.91	87.26	34.89	34.62	57.72	31.53	67.87	83.55	38.50	48.30↓4.5
w/ E-PMQ	64.23	58.48	70.75	57.56	91.28	51.50	97.55	55.27	69.69	73.64	38.42	88.39	94.25	69.68	**70.05**↑17.3
TIES-Merging	62.20	54.60	65.30	63.00	65.70	63.90	92.60	49.90	58.20	77.10	54.90	81.40	94.80	62.40	67.60↑0.0
w/ RTN	58.21	45.94	61.76	54.37	62.85	57.24	90.95	45.90	50.54	52.49	35.66	82.77	93.88	58.41	60.78↓6.8
w/ GPTQ	60.51	53.17	64.70	58.15	65.62	57.85	92.29	49.31	55.60	59.66	31.36	84.11	94.45	61.60	63.46↓4.1
w/ AWQ	58.42	48.09	61.27	52.52	64.59	56.48	91.10	46.17	50.20	51.56	37.73	82.56	93.45	57.01	60.80↓6.8
w/ E-PMQ	65.28	63.26	75.78	61.30	91.17	53.36	97.70	57.34	72.37	74.39	38.80	90.19	94.59	73.34	**72.06**↑4.5
WUDI-Merging	65.70	64.80	77.00	89.10	91.30	91.50	99.00	60.70	63.80	85.50	64.20	86.20	96.10	66.60	**78.70**↑0.0
w/ RTN	60.72	57.80	70.16	81.63	90.26	76.03	98.76	53.46	54.92	82.48	38.31	81.90	94.70	60.00	71.51↓7.2
w/ GPTQ	63.77	62.77	77.52	85.19	91.05	77.57	99.03	58.14	61.26	85.06	37.52	85.09	95.66	65.33	74.64↓4.1
w/ AWQ	61.75	60.09	70.76	79.37	90.45	77.09	98.84	55.59	55.68	80.97	37.81	81.06	94.95	61.67	71.86↓6.8
w/ E-PMQ	67.44	65.63	82.83	87.56	92.43	76.18	98.97	63.46	70.37	86.57	39.24	88.93	96.88	73.63	77.86↓0.8

Table 16: Full 14-task CLIP-ViT-L/14 results under 4-bit PMQ. All numbers are top-1 accuracy (%). Gray arrows in Avg. show changes over the corresponding full-precision merged checkpoints.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Flowers	PCAM	FER	Pets	STL10	C100	Avg.
Simple Averaging	71.20	79.00	78.70	80.40	71.30	64.60	94.30	58.70	81.90	74.20	54.80	94.60	99.30	82.40	77.50↑0.0
w/ RTN	70.33	77.95	78.03	73.85	66.28	62.80	94.33	58.40	80.27	65.03	33.91	94.55	99.13	81.24	74.01↓3.5
w/ GPTQ	70.67	77.74	77.89	76.33	70.44	62.28	94.44	58.83	80.71	75.09	37.85	94.88	99.26	81.86	75.59↓1.9
w/ AWQ	66.16	62.32	64.25	52.59	44.29	46.40	83.53	53.35	74.01	57.20	35.87	92.45	96.71	68.24	64.10↓13.4
w/ E-PMQ	72.95	84.48	84.11	78.11	96.11	71.78	94.25	67.34	94.68	84.05	41.42	95.42	98.68	85.82	**82.09**↑4.6
Task Arithmetic	60.60	53.20	48.10	53.00	50.10	54.20	93.00	41.60	59.60	75.80	53.90	89.30	94.20	57.20	63.10↑0.0
w/ RTN	58.54	49.79	46.97	45.52	44.14	46.40	92.08	41.91	57.68	50.02	39.97	88.44	93.51	54.60	57.83↓5.3
w/ GPTQ	59.88	52.37	48.90	46.52	49.55	47.44	92.84	42.45	58.64	50.03	40.26	89.02	94.06	57.10	59.22↓3.9
w/ AWQ	46.45	27.92	36.05	35.89	31.87	35.82	83.58	35.96	50.07	50.02	38.66	79.89	88.51	46.02	49.05↓14.1
w/ E-PMQ	72.75	84.85	84.75	80.52	95.54	71.87	98.63	66.76	93.71	78.61	40.42	95.48	98.61	84.86	**81.95**↑18.9
TIES-Merging	72.00	75.60	76.50	69.70	77.20	75.10	96.60	57.80	79.60	78.20	59.90	94.70	98.40	77.70	77.80↑0.0
w/ RTN	71.03	73.22	74.95	61.07	73.39	66.94	96.26	57.29	78.29	56.34	40.50	94.14	98.30	75.97	72.69↓5.1
w/ GPTQ	71.46	74.88	76.56	61.81	76.48	67.70	96.58	58.35	78.39	64.26	40.61	93.89	98.44	76.92	74.02↓3.8
w/ AWQ	67.71	60.10	68.84	47.48	62.32	57.12	95.49	54.26	76.55	57.49	41.21	92.53	97.21	71.99	67.88↓9.9
w/ E-PMQ	73.06	84.96	84.95	80.96	95.81	72.22	98.74	67.82	93.84	80.98	41.50	95.72	98.65	85.84	**82.50**↑4.7
WUDI-Merging	76.70	87.60	90.40	95.40	94.80	95.70	99.20	71.40	95.40	86.70	70.70	96.20	99.10	84.50	**88.80**↑0.0
w/ RTN	75.71	86.36	89.51	92.63	93.93	85.24	99.13	70.11	94.08	84.28	40.89	95.75	98.93	83.02	84.97↓3.8
w/ GPTQ	76.11	86.25	90.10	94.19	94.56	85.84	99.16	71.65	94.91	85.01	41.01	95.69	99.03	83.42	85.49↓3.3
w/ AWQ	72.64	76.05	84.43	84.78	91.49	80.10	98.53	66.01	91.54	78.54	40.26	94.55	98.19	79.45	81.18↓7.6
w/ E-PMQ	77.80	89.27	92.70	95.56	95.77	87.05	99.19	75.27	96.31	81.14	41.39	96.02	99.23	86.45	86.65↓2.2

E.3 Full 20-Task Results

Table 17: Full 20-task CLIP-ViT-B/32 results under 4-bit PMQ. All numbers are top-1 accuracy (%). Gray arrows in Avg. show changes over the corresponding full-precision merged checkpoints.

Method	SUN	Cars	RES	Euro	SVHN	GTSRB	MNIST	DTD	Flwr	PCAM	FER	Pets	STL	C100	C10	Food	FMNIST	EMNIST	KMNIST	SST2	Avg.
Simple Averaging	64.20	59.60	64.80	60.90	47.30	43.10	71.80	46.40	66.50	63.90	50.20	84.10	97.00	69.80	92.70	80.40	71.30	15.00	11.50	61.80	61.10↑0.0
w/ RTN	60.95	53.12	59.24	46.26	48.30	40.59	68.96	42.29	59.47	51.07	39.82	85.83	96.35	64.01	90.03	76.90	71.32	16.15	13.05	56.89	57.03↓4.1
w/ GPTQ	62.79	57.07	64.35	54.11	47.35	42.76	71.31	46.91	64.30	57.65	38.09	88.01	96.53	69.19	91.49	80.22	68.72	16.36	12.76	58.59	59.43↓1.7
w/ AWQ	60.63	50.58	57.68	43.70	48.11	42.42	69.36	43.94	59.13	53.55	40.29	85.61	95.94	64.01	89.62	76.10	65.16	15.21	13.52	56.84	56.57↓4.5
w/ E-PMQ	64.59	60.63	73.95	60.52	87.93	53.44	93.78	56.33	72.26	73.57	40.39	90.22	95.25	72.42	94.58	82.45	85.80	28.36	28.17	65.57	**69.01**↑7.9
Task Arithmetic	20.40	12.20	25.60	25.60	30.90	29.80	78.00	22.30	21.10	53.20	34.30	42.40	71.00	29.50	64.10	15.10	67.00	17.00	15.40	51.20	36.30↑0.0
w/ RTN	17.68	8.57	21.59	21.07	29.36	27.51	74.59	20.96	16.69	49.10	26.85	43.34	67.96	27.32	56.98	14.34	58.20	15.59	12.37	52.22	33.11↓3.2
w/ GPTQ	19.83	11.65	24.52	23.52	30.83	27.19	77.07	21.60	19.99	49.62	23.04	46.61	70.21	29.30	60.88	15.09	63.87	16.56	12.75	55.02	34.96↓1.3
w/ AWQ	17.72	9.66	22.87	23.07	29.54	27.58	76.21	21.44	18.47	52.62	24.92	43.80	69.26	27.41	59.15	13.29	61.93	15.59	12.81	52.28	33.98↓2.3
w/ E-PMQ	61.83	52.39	65.43	50.00	86.61	46.66	91.45	51.44	64.77	66.98	37.92	86.45	93.44	66.36	92.00	73.55	81.74	26.01	23.37	64.74	**64.16**↑27.9
TIES-Merging	51.00	36.20	47.80	45.10	58.20	57.70	92.10	40.60	44.80	66.90	47.30	73.10	89.90	51.30	86.30	50.10	76.50	21.00	19.70	55.90	55.60↑0.0
w/ RTN	48.24	30.15	45.37	36.85	55.68	47.81	90.96	36.65	40.14	50.61	32.82	74.90	89.53	49.23	84.02	46.90	71.65	19.85	16.00	53.43	51.04↓4.6
w/ GPTQ	49.96	35.22	47.62	42.19	58.39	50.53	91.88	40.48	43.03	57.45	25.23	76.42	89.93	50.63	84.62	48.66	75.89	20.30	17.02	56.18	53.08↓2.5
w/ AWQ	47.50	31.75	45.48	42.89	57.11	49.74	91.72	38.09	38.88	52.19	29.83	74.19	89.35	47.98	83.04	44.11	73.65	20.39	16.73	53.82	51.42↓4.2
w/ E-PMQ	64.04	58.85	72.57	57.89	87.70	51.37	93.12	54.52	70.45	73.08	39.51	89.13	95.10	71.29	94.20	79.97	84.77	27.12	25.94	65.79	**67.82**↑12.2
WUDI-Merging	55.10	44.80	59.30	78.50	79.70	82.90	98.10	50.30	49.30	82.00	58.50	77.70	93.40	59.80	90.30	53.10	83.90	35.40	40.00	69.00	67.10↑0.0
w/ RTN	49.86	37.50	53.76	69.70	78.45	69.52	97.57	44.04	40.62	72.16	36.67	75.44	92.20	55.83	87.85	45.74	84.61	31.79	21.17	67.49	60.60↓6.5
w/ GPTQ	52.84	42.31	58.89	75.11	78.65	71.84	98.02	47.82	47.29	77.44	36.26	79.04	92.75	58.51	88.79	49.52	84.50	33.87	23.93	67.76	63.26↓3.8
w/ AWQ	51.75	37.56	53.19	67.26	79.73	69.96	97.83	45.11	43.50	67.75	36.28	75.25	92.01	56.01	88.28	45.89	84.88	31.59	22.06	67.00	60.64↓6.5
w/ E-PMQ	64.37	57.41	79.08	85.44	89.25	73.90	98.02	56.91	61.13	85.90	38.98	87.24	96.30	70.85	94.37	72.08	87.22	36.78	29.32	69.52	**71.70**↑4.6

Table 18: Full 20-task CLIP-ViT-L/14 results under 4-bit PMQ. All numbers are top-1 accuracy (%). Gray arrows in Avg. show changes over the corresponding full-precision merged checkpoints.

Method	SUN	Cars	RES	Euro	SVHN	GTSRB	MNIST	DTD	Flwr	PCAM	FER	Pets	STL	C100	C10	Food	FMNIST	EMNIST	KMNIST	SST2	Avg.
Simple Averaging	70.70	77.70	76.40	75.30	69.50	62.10	93.70	57.70	80.80	73.60	52.70	94.20	99.20	81.70	97.00	90.70	77.40	16.10	10.40	66.10	71.10↑0.0
w/ RTN	69.90	76.84	76.10	69.74	65.49	60.88	93.61	57.07	79.10	59.23	33.51	94.14	99.04	80.82	96.83	91.82	74.09	15.61	9.22	65.02	68.40↓2.7
w/ GPTQ	70.32	76.59	76.46	71.33	68.28	60.59	93.14	58.03	79.90	72.97	35.51	94.66	99.16	81.31	96.88	92.18	74.46	16.16	10.55	68.81	69.86↓1.2
w/ AWQ	66.07	63.61	62.30	47.15	48.40	46.14	86.97	52.71	73.28	54.83	36.15	92.31	96.98	70.23	93.34	83.94	63.21	12.17	10.84	65.07	61.29↓9.8
w/ E-PMQ	72.12	82.74	83.48	79.67	94.50	71.14	97.93	65.16	92.31	80.59	40.90	95.31	98.83	84.11	98.16	92.68	90.27	49.53	13.02	70.40	**77.64**↑6.5
Task Arithmetic	17.70	17.40	45.90	89.90	59.50	59.30	96.10	36.60	41.30	81.70	49.10	65.70	81.80	28.00	76.50	24.30	80.50	62.10	73.30	57.70	57.20↑0.0
w/ RTN	21.65	11.91	17.94	25.00	18.75	23.87	76.87	22.45	21.17	50.02	27.79	55.82	73.36	23.19	61.26	16.08	53.85	12.46	10.50	50.19	33.71↓23.5
w/ GPTQ	22.64	13.72	18.81	26.30	19.21	24.79	78.34	22.45	21.56	50.02	29.51	57.26	75.83	25.32	64.81	16.34	56.88	12.07	10.63	50.36	34.84↓22.4
w/ AWQ	12.23	5.43	13.08	24.30	13.29	19.83	63.04	18.88	15.76	50.02	26.33	38.81	66.45	16.15	48.13	10.49	39.91	10.78	10.21	50.14	27.66↓29.5
w/ E-PMQ	72.06	83.15	82.75	76.63	93.50	70.18	97.80	64.57	91.97	73.44	40.47	95.31	98.71	82.76	98.01	91.91	89.91	48.57	12.44	70.02	**76.71**↑19.5
TIES-Merging	64.40	56.60	49.10	42.10	67.20	56.80	95.30	46.20	64.90	78.30	54.90	91.30	95.70	62.90	92.90	70.50	82.00	19.90	10.90	56.90	63.00↑0.0
w/ RTN	63.03	53.95	48.51	40.48	63.77	51.18	94.76	46.81	63.07	50.03	38.14	90.98	95.30	61.95	92.00	69.22	81.65	17.34	10.01	73.31	60.27↓2.7
w/ GPTQ	63.89	55.34	48.97	40.19	66.72	51.81	95.36	47.87	65.56	50.09	38.56	90.76	95.60	62.88	92.56	70.72	81.56	17.94	9.88	75.34	61.08↓1.9
w/ AWQ	60.02	44.10	44.32	33.59	60.25	45.90	95.88	45.37	64.42	50.08	39.37	87.95	94.46	60.35	90.73	64.43	77.84	19.09	11.07	71.61	58.04↓5.0
w/ E-PMQ	71.99	82.89	84.06	80.00	94.51	71.10	97.90	65.59	92.36	78.20	40.47	95.39	98.73	83.88	98.19	92.56	90.35	47.65	13.13	70.07	**77.45**↑14.5
WUDI-Merging	70.30	72.10	73.30	69.70	81.90	84.30	98.10	56.80	85.90	83.70	64.20	94.50	97.50	72.90	95.70	83.50	89.60	33.20	33.30	74.80	75.80↑0.0
w/ RTN	69.19	68.75	71.27	62.07	80.09	72.41	98.05	55.59	83.67	74.26	38.92	94.17	97.25	71.49	95.32	80.92	89.64	30.03	13.35	78.36	71.24↓4.6
w/ GPTQ	69.77	70.43	73.49	63.78	81.69	73.64	98.11	57.55	84.88	76.66	39.20	94.03	97.29	72.45	95.53	82.37	89.98	30.90	13.39	78.64	72.19↓3.6
w/ AWQ	66.49	57.02	68.78	55.52	78.56	66.70	98.19	56.12	81.64	74.29	39.36	92.61	96.78	67.36	94.22	77.79	86.65	30.93	14.12	78.09	69.06↓6.7
w/ E-PMQ	76.61	84.74	90.65	94.48	93.96	84.35	98.73	70.74	94.99	82.87	40.44	95.80	99.20	84.86	98.18	91.17	91.98	48.37	17.02	79.63	**80.94**↑5.1

Table 17 reports the full 20-task CLIP-ViT-B/32 results. This setting is more challenging for naive PMQ, and E-PMQ provides large improvements by using expert-guided targets during quantization.

Table 18 reports the full 20-task CLIP-ViT-L/14 results. E-PMQ again outperforms naive PMQ baselines across the main merging methods, confirming its scalability to larger task suites.

F LLM Implementation Details

For the Llama experiments, we construct merged models using Task Arithmetic with the merge coefficient set to 0.3. The source experts correspond to four capability-oriented tasks: instruction following, coding, mathematics, and multilingual understanding. We evaluate Llama-3.2-3B-Instruct and Llama-3.1-8B-Instruct merged models.

All quantized LLMs use 4-bit weight-only quantization with group size 32, calibration batch size 4, and a maximum calibration sequence length of 512. The calibration set of each expert contains 256 samples; GPTQ and AWQ use the pooled set. The saved quantized weights use bfloat16 as the storage dtype for non-quantized tensors. We compare E-PMQ with AWQ and GPTQ, and also report the full-precision merged model without quantization. For AWQ, we set the grid-search parameter to 20 in the reported baseline experiments. E-PMQ uses the same calibration data and follows the same forward-order layer-wise quantization protocol as the other PMQ methods.

For E-PMQ, the expert model paths are instantiated from the corresponding model family and task name. We set the global anchor scaling hyperparameter to $\alpha = 1$ for the LLM experiments. Source experts are used only during calibration to construct expert-guided targets. After quantization, the deployed model is a single 4-bit merged model without additional experts or inference-time modules.

G Diagnostic Analysis of PMQ

The main results show that E-PMQ consistently improves over direct post-merge quantization. We provide two diagnostic analyses to better understand this improvement. First, we compare E-PMQ with a natural expert-aware pipeline, Quantize-then-Merge, to verify that the gain does not simply come from accessing source experts. Second, Figure 3 in the main text measures the accumulated residual $\|QX_i - W_iX_i\|_F^2 = \|D_{\text{quant}} + D_{\text{merge}}\|_F^2$, which E-PMQ reduces relative to GPTQ without claiming that either term shrinks on its own.

Comparison with Quantize-then-Merge. A natural expert-aware alternative is to first quantize each source expert and then merge the quantized experts. We denote this pipeline as Quantize-then-Merge. As shown in Table 19, this baseline performs worse than both GPTQ and E-PMQ, suggesting that independently quantizing experts can disrupt the parameter differences used by the merging algorithm. In contrast, E-PMQ keeps the full-precision merged model as the starting point and uses source experts only during post-merge quantization.

Table 19: Comparison with Quantize-then-Merge on CLIP-ViT-B/32 under Task Arithmetic.

Method	Avg.
Task Arithmetic	68.00
w/ GPTQ	65.03
Quantize-then-Merge	56.06
w/ E-PMQ	**73.61**

Table 20: Anchor-strength ablation on CLIP-ViT-B/32 8-task setting under 4-bit PMQ. All numbers are top-1 accuracy (%). Gray arrows in Avg. show changes over the corresponding full-precision merged checkpoints.

Method	α	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Task Arithmetic	–	57.10	55.70	64.90	76.70	77.90	68.50	96.10	47.20	68.00$_{\uparrow 0.0}$
w/ GPTQ	–	55.60	53.54	63.83	69.11	77.87	56.99	95.88	47.45	65.03$_{\downarrow 3.0}$
	0	0.29	0.51	2.17	16.41	8.72	2.39	9.78	2.71	5.37$_{\downarrow 62.6}$
	0.01	67.03	64.36	78.52	66.30	94.75	57.03	98.92	61.97	73.61$_{\uparrow 5.6}$
w/ E-PMQ	0.1	67.09	64.03	79.24	68.33	94.55	58.81	98.72	61.91	**74.09**$_{\uparrow 6.1}$
	1	66.12	63.52	78.37	70.78	92.71	61.27	98.31	60.48	73.94$_{\uparrow 5.9}$
	10	64.60	60.66	76.37	76.37	89.92	64.37	97.59	55.80	73.21$_{\uparrow 5.2}$
TIES-Merging	–	67.10	64.20	74.10	76.80	77.70	69.40	94.10	54.00	72.20$_{\uparrow 0.0}$
w/ GPTQ	–	65.60	61.10	72.30	67.70	76.70	62.90	93.90	53.00	69.10$_{\downarrow 3.1}$
	0	0.29	0.52	2.79	9.81	8.90	3.28	9.74	1.22	4.57$_{\downarrow 67.6}$
	0.01	67.61	66.65	80.46	67.19	94.66	59.20	98.96	63.24	**74.75**$_{\uparrow 2.6}$
w/ E-PMQ	0.1	67.47	66.40	80.05	67.67	93.98	61.31	98.65	62.13	74.71$_{\uparrow 2.5}$
	1	66.72	65.02	78.24	69.33	91.71	60.71	97.71	60.32	73.72$_{\uparrow 1.5}$
	10	65.14	60.75	75.78	71.04	86.89	63.25	96.25	55.80	71.86$_{\downarrow 0.3}$

Expert-relative output error. Figure 3 in the main text reports the layer-wise expert-relative output error. Together with Table 19, it indicates that the gain of E-PMQ does not merely come from accessing source experts, but from using them after merging to reduce accumulated expert-relative deviation.

H Full Anchor-Strength Ablation

Table 20 reports the full per-task anchor-strength ablation. Removing the anchor by setting $\alpha = 0$ causes severe collapse, while positive anchor strengths remain stable and consistently outperform GPTQ in average accuracy. This supports the necessity of merged-weight anchoring in the E-PMQ objective.

I Bit-Width Analysis

Figure 4 compares E-PMQ with GPTQ under different bit-widths. E-PMQ consistently outperforms GPTQ from 3-bit to 8-bit under both Task Arithmetic and TIES-Merging. The largest gains appear in the more aggressive low-bit regimes, where quantization deviation is more severe and naive reconstruction of the merged model is least reliable.

J Quantization Cost and Calibration Budget

We measure the quantization-stage cost of accessing source experts. This extra cost is incurred only during quantization. After quantization, E-PMQ has the same single-model inference form, parameter count, and bit-width as the corresponding GPTQ baseline.

Accessing source experts during calibration does not require all experts to reside on the GPU at once. We compare two loading strategies: *All-at-once*, which keeps every expert resident, and *Sequential*, which processes one expert at a time and caches its layer-wise calibration statistics. Table 21 profiles W4 E-PMQ on an NVIDIA A800 using CLIP-ViT-B/32 and the first $2/4/6/8$ experts from the eight-

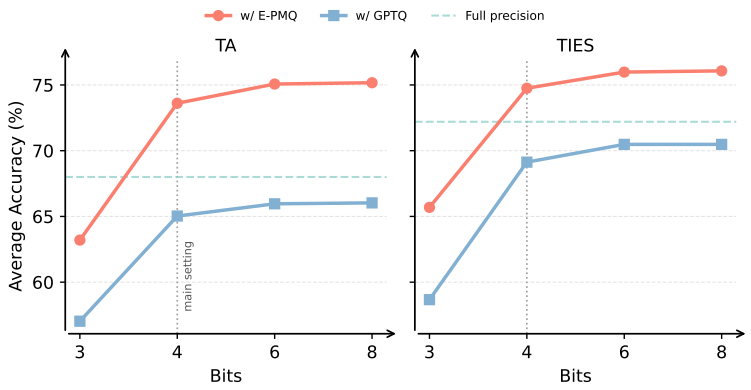


Figure 4: Bit-width analysis on CLIP-ViT-B/32. E-PMQ consistently outperforms GPTQ from 3-bit to 8-bit.

task suite under Task Arithmetic, with group size 128, 256 calibration samples per expert, and a calibration batch size of 32.

Table 21: GPU-loading cost of W4 E-PMQ on CLIP-ViT-B/32 as the number of experts increases. Profiling is conducted on an NVIDIA A800.

Experts	Loading	Time (s)	Peak GPU alloc. (GiB)	Peak GPU reserved (GiB)	Peak CPU (GiB)	Source ckpt	Temp. cache (GiB)
2	All-at-once	27.70	3.73	4.14	2.43	667.4 MiB	0
2	Sequential	36.34	2.33	2.56	2.19	667.4 MiB	1.59
4	All-at-once	33.34	5.72	6.20	2.79	1.30 GiB	0
4	Sequential	48.62	2.37	2.62	2.41	1.30 GiB	3.18
6	All-at-once	40.08	7.71	8.25	3.09	1.96 GiB	0
6	Sequential	62.49	2.38	2.63	2.42	1.96 GiB	4.77
8	All-at-once	53.84	9.69	10.30	3.42	2.61 GiB	0
8	Sequential	76.65	2.38	2.65	2.52	2.61 GiB	6.36

Sequential loading reduces peak GPU memory from $O(K)$ to approximately $O(1)$. With eight experts, peak GPU allocation falls by 75.4%, while calibration time rises by 42.4% and the temporary disk cache grows linearly. The loading strategy can therefore be chosen according to the available resources. After calibration, deployment still requires only one W4 model, independent of the number of experts.

We further profile complete pipelines on the same eight-expert CLIP setting and hardware. Table 22 reports total processing time from loading the required checkpoints to saving the final quantized model. It includes method-specific calibration, representation repair where applicable, and intermediate-checkpoint I/O, but excludes final evaluation. The common 2.61 GiB source-checkpoint storage is reported separately and excluded from method-specific temporary storage. Accuracy comparisons with HDRQ and Surgery + GPTQ are provided in App. B.

Table 22: Complete-pipeline cost on eight-expert CLIP-ViT-B/32 under W4A16. Total process time is measured from loading the required checkpoints to saving the final quantized model, excluding evaluation.

Method	Time (s)	Speedup vs. HDRQ	Peak GPU (GiB)	Peak CPU (GiB)	Final model	Temp. storage
E-PMQ All-at-once	53.84	165.72×	9.69	3.42	293.54 MiB	0
E-PMQ Sequential	76.65	116.41×	2.38	2.52	293.54 MiB	6.36 GiB
Surgery + GPTQ	142.24	62.73×	1.19	6.38	293.89 MiB	≈0
HDRQ (W4A16)	8,922.55	1.00×	1.41	6.28	293.54 MiB	2.71 GiB

The profiling results support three conclusions. First, E-PMQ is substantially faster because it calibrates the final merged model once, whereas Surgery + GPTQ uses two stages and HDRQ repeats a long layer-wise optimization for each expert. Second, E-PMQ offers a configurable memory–storage

trade-off: All-at-once loading is fastest but uses more GPU memory, while Sequential loading reduces peak GPU allocation to 2.38 GiB at the cost of 6.36 GiB of temporary storage. Third, after quantization, all compared methods produce a comparably small W4 model; E-PMQ does so without a residual adapter and with much lower processing time than HDRQ.

We also examine the trade-off between calibration budget, quantization-stage computation, and final 4-bit accuracy in Table 23. The calibration budget is measured as samples per task; in the 8-task setting, n samples per task correspond to $8n$ total calibration images. With only 64 samples per task, E-PMQ reaches 72.23 average accuracy, outperforming GPTQ with 256 samples per task by 7.20 points.

This result suggests that expert-guided targets provide a more informative calibration signal than merged-model reconstruction alone. E-PMQ therefore trades additional pre-deployment computation for better use of limited calibration data and stronger low-bit merged-model quality.

Table 23: Calibration budget and quantization time.

Method	Samp.	Avg.	Time	Ratio	Samp.	Avg.	Time	Ratio	Samp.	Avg.	Time	Ratio
Task Arithmetic	–	68.00	–	–	–	68.00	–	–	–	68.00	–	–
w/ GPTQ	64	64.86	29.2s	1.00×	128	64.77	33.9s	1.00×	256	65.03	71.0s	1.00×
w/ E-PMQ	64	**72.23**	52.8s	1.81×	128	**73.58**	101.4s	2.99×	256	**73.61**	172.2s	2.43×

K Implementation Details

Unless otherwise stated, CLIP and FLAN-T5 experiments use weight-only quantization with group size 128. The calibration batch size is 32 and the evaluation batch size is 128. These FusionBench experiments use 256 calibration samples per task; for a K-task merged model, this corresponds to $256K$ total calibration samples. LLM settings differ and are given in App. F.

For E-PMQ, X_i^ℓ is collected from the current quantization trajectory of the partially quantized merged model on the calibration subset associated with expert/task i. The expert-guided target is $W_i^\ell X_i^\ell$: expert weights on merged-path activations. We do not run the expert models to collect activations.

We set $\alpha = 0.01$ for Simple Averaging, Task Arithmetic, and TIES-Merging, and $\alpha = 10$ for WUDI-Merging. For bit-width analysis, we keep the same calibration protocol and vary only the weight bit-width. For the calibration-budget analysis in App. J, we vary the number of calibration samples per task while keeping the remaining quantization settings fixed. Test sets are used only for final evaluation.