# DISTILL GLOBALLY, TRAIN LOCALLY: ERROR-LOCALIZED RECOVERY FOR 4-BIT LLMS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Low-bit quantization enables efficient large language model (LLM) inference, but maintaining accuracy with 4-bit weights and activations (W4A4) remains challenging. Local calibration in post-training quantization (PTQ) lacks direct supervision of final predictions, limiting further accuracy recovery. Full-model quantization-aware distillation (QAD) can recover near-BF16 accuracy through global supervision, but requires costly model-wide optimization. To preserve full-model QAD accuracy while minimizing training cost, we propose LoDiQ, a lightweight error-localized distillation framework with three components: (1) propagated-error localization identifies sensitive input channels, (2) quantized residual compensation strengthens recovery within expanded low-bit GEMMs, with Gradient-Guided Residual Compensation Allocation (GRCA) supporting lower-budget configurations, and (3) localized distillation trains only the selected corrective weights using compact teacher targets, avoiding full-vocabulary teacher-logit storage. Experiments across the Qwen and Llama model families span language modeling, zero-shot commonsense reasoning, broad-domain knowledge, and mathematical reasoning. LoDiQ surpasses state-of-the-art PTQ baselines in downstream accuracy under both MXFP4 and NVFP4 W4A4 quantization. On a matched reasoning evaluation, LoDiQ outperforms full-model QAD while using only 1/100 as many distillation tokens and updating only 4.67% as many parameters, substantially reducing optimization workload and trainable-state memory. Inference benchmarks on RTX 5090 show up to 3.20× and 3.14× time-to-first-token (TTFT) speedups over BF16 in MXFP4 and NVFP4, respectively.

## 1 INTRODUCTION

The substantial memory and computation requirements of large language models (LLMs) create a growing demand for efficient deployment. Fine-grained numerical formats such as MXFP4 and NVFP4 have been proposed to make practical low-bit LLM inference possible on modern hardware through efficient low-precision matrix multiplication (Rouhani et al., 2023; Egiazarian et al., 2026). However, the accuracy degradation introduced by low-bit quantization remains a central challenge.

Several effective post-training quantization (PTQ) approaches have been developed to mitigate this problem (Nagel et al., 2021). Quantization preprocessing and reconstruction reduce error through scaling, rotations, and calibration-driven optimization, as exemplified by SmoothQuant, QuaRot, GPTQ, CBQ, and MR-GPTQ (Xiao et al., 2023; Ashkboos et al., 2024; Frantar et al., 2023; Ding et al., 2025; Egiazarian et al., 2026). Structured compensation methods allocate additional correction capacity to quantization-sensitive channels, as exemplified by SERQ and ARCQuant (Park et al., 2026; Meng et al., 2026). These representative approaches rely on calibration statistics, intermediate reconstruction objectives, or analytical transformations rather than direct supervision of the final predictive distribution, limiting further accuracy recovery at low precision.

Quantization-aware distillation (QAD) directly addresses the missing model-level signal by minimizing the Kullback–Leibler (KL) divergence between the predictive distributions of a quantized student and a full-precision teacher. Recent QAD results consistently recover quantized models to near-BF16 accuracy (Xin et al., 2026). Full-model QAD, however, requires model-wide gradients and optimizer states, backpropagates through the entire student, and can require substantial distilla-



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.

tion data. This accuracy–cost trade-off raises a central question: can model-level distillation retain near-BF16 recovery while limiting optimization to error-sensitive parameters?

To answer this question, we propose LoDiQ, a lightweight error-localized distillation framework for 4-bit LLM quantization. LoDiQ introduces a new operating point between locally calibrated PTQ and model-wide QAD: teacher supervision remains defined at the final predictive distribution, but trainable recovery capacity is concentrated in quantization-sensitive channel weights. Channel localization defines the trainable weight subset, and quantized residual compensation augments its recovery capacity. Budgeted allocation provides lower-cost variants, while compact teacher targets remove the full-vocabulary cache bottleneck. Only the corresponding corrective weights are optimized and the remaining parameters stay frozen. After distillation, the learned correction is quantized and packed with the frozen weights into expanded low-bit linear operations. LoDiQ thereby retains model-level supervision with a small, structured trainable state and a uniform target-format inference representation. Our contributions are summarized as follows:

- We propose LoDiQ, an error-localized distillation framework for 4-bit LLM quantization. To the best of our knowledge, LoDiQ is the first framework to restrict parameter updates under model-level distillation to a structured subset of quantization-sensitive weight columns for 4-bit LLM recovery. Teacher supervision remains defined at the final predictive distribution, while only the localized corrective weights are optimized. The learned correction remains in the target 4-bit representation for packed low-bit inference.
- We introduce propagated-error channel localization to identify the trainable weight columns. Quantized residual compensation further improves recovery with trainable expanded weights. Gradient-Guided Residual Compensation Allocation (GRCA) further supports lower-budget configurations by allocating residual compensation across layers under a global parameter budget. Compact KL targets retain top-16 token probabilities and aggregate vocabulary-remainder mass, reducing the cache for one million supervised tokens from 304.48 GB to 0.098 GB, a reduction of more than 3,100$\times$.
- We conduct controlled comparisons with format-matched PTQ baselines and full-model QAD across representative LLMs and hardware-native W4A4 formats. Beyond downstream accuracy, the evaluation examines training cost, sensitivity to distillation data and token budget, compact-target fidelity, and realized inference efficiency. LoDiQ consistently improves the accuracy–training-cost trade-off over strong PTQ and QAD references while preserving efficient low-bit inference.

## 2 Background and Motivation

### 2.1 Accuracy challenges in 4-bit LLM quantization

MXFP4 and NVFP4 use fine-grained block scaling to better match local group ranges, improving effective quantization resolution. MXFP4 uses an E8M0 scale for each block of 32 E2M1 values, while NVFP4 uses an E4M3 scale for 16 E2M1 values with an additional per-tensor scale (Open Compute Project, 2023; Alvarez et al., 2025). However, activation outliers can still induce substantial quantization error, limiting accuracy under W4A4 quantization (Egiazarian et al., 2026). Classical distribution-shaping strategies based on equivalent scaling or global rotation do not transfer cleanly to fine-grained W4A4 formats. SmoothQuant transfers activation dynamic range to the weights through equivalent scaling. Although effective at INT8, this transfer can aggravate weight quantization under W4A4 (Xiao et al., 2023; Meng et al., 2026). Rotation-based methods such as QuaRot, SpinQuant, and DuQuant redistribute outliers across dimensions through orthogonal transformations (Ashkboos et al., 2024; Liu et al., 2025b; Lin et al., 2024). Global rotations spanning microscaling groups can spread outliers across blocks and weaken block isolation (Meng et al., 2026).

Format-aware reconstruction and compensation provide a complementary route. MR-GPTQ combines group-wise Hadamard transforms and static reordering with format-aware GPTQ and scale optimization (Egiazarian et al., 2026). ARCQuant adds quantized residual components to selected activation channels, whereas SERQ uses activation flattening and a saliency-guided low-rank matrix to reconstruct quantization error over salient weight rows (Meng et al., 2026; Park et al., 2026). Despite their different mechanisms, these methods derive recovery decisions from calibration statistics

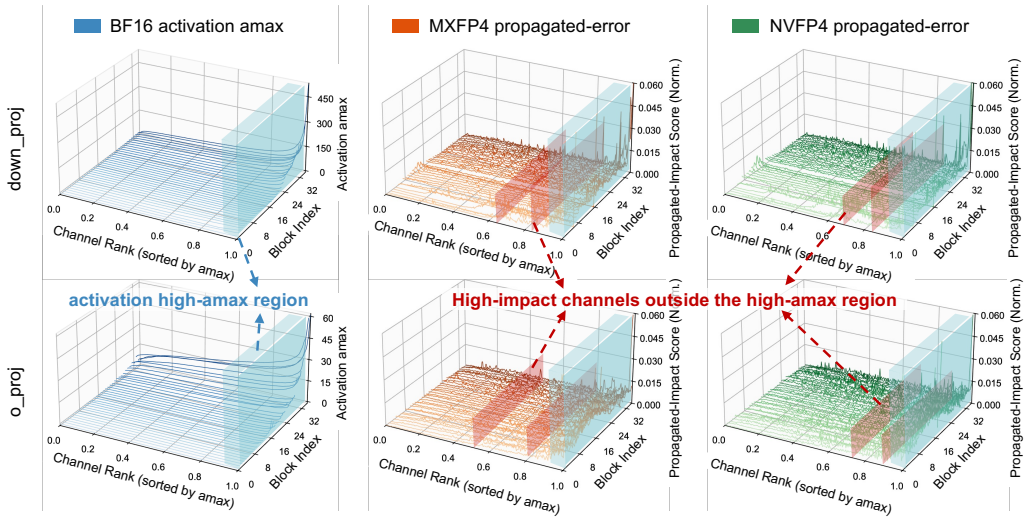


Figure 1: Channel-wise propagated-impact scores in Qwen3-8B. The scores concentrate in a small subset of channels and exhibit distinct profiles across projections and blocks.

or intermediate reconstruction rather than the final predictive distribution. They form the principal PTQ baselines in our evaluation.

### 2.2 Model-level supervision for quantization recovery

QAD directly targets quantization recovery by matching the final teacher and student distributions through model-level KL (Hinton et al., 2015; Xin et al., 2026):

$$\mathcal{L}_{\text{QAD}} = D_{\text{KL}}(p_T(\cdot \mid x) \,\|\, p_S(\cdot \mid x)) = \sum_{y \in \mathcal{V}} p_T(y \mid x) \log \frac{p_T(y \mid x)}{p_S(y \mid x)}, \tag{1}$$

where $p_T$ and $p_S$ are the teacher and student distributions and $\mathcal{V}$ is the vocabulary.

QAD consistently recovers near-BF16 accuracy and therefore serves as the matched training-based recovery baseline in our evaluation (Xin et al., 2026). Full-parameter QAD, however, requires model-wide gradients, optimizer states, and iterative backpropagation through the student.

Restricting the trainable state offers a natural route to lower this cost. QLoRA demonstrates parameter-efficient training over a frozen weight-only 4-bit backbone by optimizing high-precision low-rank adapters (Dettmers et al., 2023). Retaining these adapters in their trained form adds high-precision narrow GEMMs to each adapted linear operation, increasing inference overhead and preventing a uniform W4A4 execution path. LoDiQ instead retains the QAD objective while restricting updates to quantization-sensitive weight columns and representing the learned correction in the target low-bit format.

Local calibration is also difficult to apply uniformly in sparse MoE models because calibration data can cover experts unevenly (Chen et al., 2025). Model-level KL follows the experts activated by each distillation sequence, so corrective updates reflect their usage and downstream influence. Coverage still depends on the representativeness of the distillation data.

### 2.3 Quantization error locality

To examine the relation between activation magnitude and its propagated quantization impact, we first order the input channels of each linear layer by their calibration-set maximum absolute activation magnitude (amax), $a_j = \max_t |(\boldsymbol{X})_{t,j}|$, and apply the same permutation to the weight columns. This ordering preserves values and the linear mapping while grouping channels with comparable ranges to improve within-group homogeneity (Meng et al., 2026). Let $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ and $\boldsymbol{W} \in \mathbb{R}^{d_{\text{out}} \times d}$ denote the reordered activations and weights for $n$ tokens, with output $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{W}^\top$.

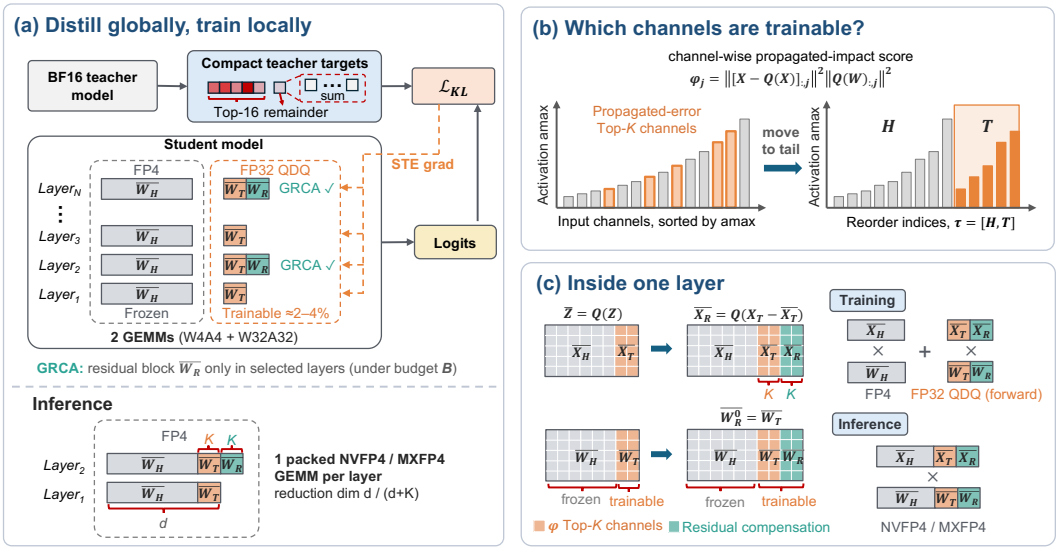


Figure 2: Overview of LoDiQ. Offline calibration localizes and reorders propagated-error-sensitive channels. Compact-target distillation optimizes channel-localized weights with residual compensation for packed low-bit inference.

We model the output perturbation propagated from activation quantization through a quantized linear projection as

$$\Delta \boldsymbol{Y} = (\boldsymbol{X} - Q(\boldsymbol{X}))\, Q(\boldsymbol{W})^\top = \sum_{j=1}^{d} [\boldsymbol{X} - Q(\boldsymbol{X})]_{:,j}\, Q(\boldsymbol{W})_{:,j}^\top. \tag{2}$$

Based on Eq. 2, we define the channel-wise propagated-impact score

$$\phi_j = \left\| [\boldsymbol{X} - Q(\boldsymbol{X})]_{:,j} \right\|_2^2 \, \| Q(\boldsymbol{W})_{:,j} \|_2^2 \,. \tag{3}$$

Figure 1 compares activation amax with normalized propagated-impact scores. Across both formats and projection types, propagated-impact scores remain low for most channels but rise sharply within a small subset, revealing pronounced channel-wise locality in propagated quantization error. However, these high-impact channels do not fully coincide with the activation-amax profiles. Activation magnitude alone does not account for the realized quantization residual or its propagation through the corresponding quantized weight column. We therefore use $\phi_j$ as the proxy for localizing sensitive channels in Section 3.1.

# 3 Method

## 3.1 Propagated-error channel localization

As illustrated in Figure 2, LoDiQ computes the propagated-impact score in Eq. 3 under the initial activation-amax ordering and selects the top-$K$ input channels from every linear projection within the W4A4 quantization boundary. The selected channels define the parameter subset optimized by localized distillation. To form a contiguous trainable region, a second stable reordering moves the selected channels to a tail $T$ and applies the same reordering to the weight columns. The frozen channels form $H$, yielding the final reorder indices $\tau = [H, T]$ while preserving the activation-amax order within each subset. Channel selection and $\tau$ are determined once and remain fixed. Subsequent grouped quantization uses the final layout $\tau$. At inference, $\tau$ is realized as an indexed channel reorder fused with activation quantization. The corresponding weight-column reorder and packing are performed offline.

For a representative linear layer, $\boldsymbol{X} = [\boldsymbol{X}_H, \boldsymbol{X}_T]$ and $\boldsymbol{W} = [\boldsymbol{W}_H, \boldsymbol{W}_T]$ now denote the activation and weight columns in the final layout $\tau$. Throughout the method, an overbar denotes target-format grouped quantization, $\bar{\boldsymbol{Z}} = Q(\boldsymbol{Z})$. Both weight blocks are initialized from the corresponding pre-trained columns. The head block $\bar{\boldsymbol{W}}_H = Q(\boldsymbol{W}_H)$ is fixed after quantization. The tail block $\boldsymbol{W}_T$ is maintained as an FP32 master parameter and optimized through model-level distillation; its quantized value $\bar{\boldsymbol{W}}_T = Q(\boldsymbol{W}_T)$ is used in the forward pass. During distillation, the frozen and trainable blocks are evaluated separately because they use different parameter-storage precisions, and their outputs are summed:

$$\hat{\boldsymbol{Y}} = \bar{\boldsymbol{X}}_H \bar{\boldsymbol{W}}_H^\top + \bar{\boldsymbol{X}}_T \bar{\boldsymbol{W}}_T^\top. \tag{4}$$

Only $\boldsymbol{W}_T$ is updated through model-level distillation. This separation is restricted to training. For inference, both blocks use the same target 4-bit format and are packed into a single reordered weight matrix. Equation 4 preserves the original reduction width, and every quantized linear layer uses this localized trainable base.

### 3.2 Gradient-guided residual compensation allocation

**Residual compensation.** The trainable tail in Eq. 4 receives only the quantized primary component $\bar{\boldsymbol{X}}_T$ and has no direct access to the discarded activation residual. LoDiQ introduces a quantized residual component to provide additional input information and recovery capacity. Before quantizing the residual, the selected-channel contribution admits the exact decomposition

$$\boldsymbol{X}_T \boldsymbol{W}_T^\top = \bar{\boldsymbol{X}}_T \boldsymbol{W}_T^\top + \boldsymbol{X}_R \boldsymbol{W}_T^\top, \qquad \boldsymbol{X}_R = \boldsymbol{X}_T - \bar{\boldsymbol{X}}_T. \tag{5}$$

LoDiQ augments the quantized primary component with $\bar{\boldsymbol{X}}_R$ and a trainable residual weight block $\boldsymbol{W}_R \in \mathbb{R}^{d_{\text{out}} \times K}$:

$$\hat{\boldsymbol{Y}} = \bar{\boldsymbol{X}}_H \bar{\boldsymbol{W}}_H^\top + \bar{\boldsymbol{X}}_T \bar{\boldsymbol{W}}_T^\top + \bar{\boldsymbol{X}}_R \bar{\boldsymbol{W}}_R^\top. \tag{6}$$

Before distillation, $\boldsymbol{W}_R$ is initialized by copying $\boldsymbol{W}_T$, giving $\overline{\boldsymbol{W}_R^0} = \bar{\boldsymbol{W}}_T$ under the same grouped quantization rule. The superscript 0 denotes initialization. The two weight blocks are not tied because $\bar{\boldsymbol{X}}_T$ and $\bar{\boldsymbol{X}}_R$ have different statistics and carry complementary information. Both matrices are maintained as FP32 master parameters and jointly optimized through model-level distillation. Since $\boldsymbol{X}_R$ and the weights are quantized in Eq. 6, the implemented forward is a trainable low-bit approximation of the exact decomposition above. Upgrading a layer from Eq. 4 to Eq. 6 adds only $\boldsymbol{W}_R$, adding $Kd_{\text{out}}$ trainable parameters and the same number of multiply–accumulate operations per token processed by the layer. Residual augmentation expands the reduction dimension from $d$ to $d + K$ but does not introduce a high-precision residual bypass.

**Budgeted allocation.** The full-budget setting adds residual compensation to every quantized linear layer, doubling the trainable parameters of the tail-only configuration. Gradient-Guided Residual Compensation Allocation (GRCA) selects which layers receive compensation under a smaller residual-parameter budget.

For a fixed layer $l$, let $\boldsymbol{y}_i$ denote its full-precision output for calibration sequence $i$, vectorized over token positions and output channels. The gradient $\boldsymbol{g}_i = \partial \mathcal{L}_{\text{NLL}} / \partial \boldsymbol{y}_i$ measures sensitivity to the model-level prediction loss, where $\mathcal{L}_{\text{NLL}}$ is the mean negative log-likelihood of the observed next tokens. Let $\boldsymbol{e}_i$ and $\boldsymbol{e}'_i$ denote the output errors without and with residual compensation, respectively, relative to $\boldsymbol{y}_i$. Both candidates use the same full-precision layer input and are evaluated before distillation. GRCA computes the per-sequence reduction in gradient-weighted squared error and the confidence-adjusted layer score as

$$\Delta_i = \|\boldsymbol{g}_i \odot \boldsymbol{e}_i\|_2^2 - \|\boldsymbol{g}_i \odot \boldsymbol{e}'_i\|_2^2, \qquad G_l = \text{mean}_i(\Delta_i) - \frac{\text{std}_i(\Delta_i)}{\sqrt{N}}. \tag{7}$$

Here $\odot$ denotes element-wise multiplication, and $N$ is the number of calibration sequences. Subtracting one standard error favors consistent reductions at prediction-sensitive output coordinates. The resulting $G_l$ is an error-reduction proxy rather than the actual change in NLL or post-distillation accuracy.

Let $\mathcal{I}$ index the quantized linear layers. Adding $\boldsymbol{W}_R$ to layer $l$ costs $c_l = Kd_{\text{out},l}$ parameters. Given a global budget $B$ for these additional parameters, the residual allocation is formulated as

$$\mathcal{A}^\star = \arg\max_{\mathcal{A} \subseteq \mathcal{I}} \sum_{l \in \mathcal{A}} G_l \quad \text{s.t.} \quad \sum_{l \in \mathcal{A}} c_l \le B. \tag{8}$$

Dynamic programming selects the layer combination with the maximum total estimated gain under the parameter budget. NLL gradients guide allocation without updating weights, and subsequent distillation uses teacher KL. Appendix A.2 provides the scoring and optimization details.

### 3.3 Error-localized distillation with compact KL targets

After GRCA determines the set $\mathcal{A}$ of residual-compensated linear layers, the set of trainable corrective weights $\Theta_{\text{train}}$ contains $\boldsymbol{W}_T$ for every quantized linear layer and $\boldsymbol{W}_R$ only for layers in $\mathcal{A}$. During distillation, the frozen low-bit weight blocks remain fixed, while the weights in $\Theta_{\text{train}}$ are maintained as FP32 master parameters. Frozen linear weights are quantized once. Trainable weights and activations are fake-quantized to the target format in each forward pass, with straight-through estimators (STE) for backpropagation. The frozen and trainable contributions are computed separately and summed, with gradients retained through the frozen path to upstream trainable parameters.

LoDiQ distills from compact teacher targets using a top-16-plus-remainder KL objective. At each supervised position, let $\mathcal{K}_x$ denote the teacher-selected top-16 token set and let $\mathcal{R}_x = \mathcal{V} \setminus \mathcal{K}_x$ denote the vocabulary remainder. Rather than cache full-vocabulary teacher logits, compact targets store the token indices in $\mathcal{K}_x$, their teacher log-probabilities, and the aggregate log-probability of $\mathcal{R}_x$. These quantities define a 17-way partition comprising 16 singleton tokens and one remainder cell. Let $\tilde{p}_T$ and $\tilde{p}_S$ denote the corresponding teacher and student probabilities. The objective is

$$\mathcal{L}_{\text{KL}} = D_{\text{KL}}(\tilde{p}_T \,\|\, \tilde{p}_S) = \sum_{y \in \mathcal{K}_x} p_T(y \mid x) \log \frac{p_T(y \mid x)}{p_S(y \mid x)} + p_T(\mathcal{R}_x \mid x) \log \frac{p_T(\mathcal{R}_x \mid x)}{p_S(\mathcal{R}_x \mid x)}. \quad (9)$$

On held-out validation data, the teacher-selected top-16 tokens account for 99.77% of the vocabulary probability mass on average. Compact targets retain the remaining mass as one aggregate cell rather than supervising its token-wise distribution. By the chain rule for KL divergence, the omitted within-remainder term is weighted by this small residual mass, while the aggregate remainder probability is still matched. The trainable corrective weights are optimized with Eq. 9. After distillation, the quantized frozen and learned weight blocks are packed into a single matrix per layer for low-bit GEMM. Residual-compensated layers use an expanded matrix without a separate residual GEMM.

## 4 Experiments

### 4.1 Setup

**Models and Tasks.** We evaluate Llama-3.2-3B (Meta, 2024), Llama-3-8B (Grattafiori et al., 2024), Qwen3-8B and Qwen3-32B (Yang et al., 2025), and Moonlight-16B-A3B-Instruct (Liu et al., 2025a). To cover language modeling, broad-domain knowledge, and downstream reasoning, we report WikiText-2 perplexity (Merity et al., 2017), five-shot MMLU (Hendrycks et al., 2021), and the mean zero-shot accuracy over six benchmarks: ARC-Challenge and ARC-Easy (Clark et al., 2018), HellaSwag (Zellers et al., 2019), LAMBADA (Paperno et al., 2016), PIQA (Bisk et al., 2020), and WinoGrande (Sakaguchi et al., 2020). Additional reasoning evaluations include GPQA-Diamond (Rein et al., 2024) and five-shot GSM8K (Cobbe et al., 2021). Following the model–benchmark pairing used in QAD (Xin et al., 2026), we separately evaluate AceReason-Nemotron-1.1-7B (Liu et al., 2026) on AIME25, with AIME26 included as a more recent extension (Zhang & Math-AI Team, 2025; 2026).

**Baselines.** For representative and format-matched evaluation, we compare LoDiQ with BF16, direct round-to-nearest quantization (RTN), and strong recent PTQ baselines. MR-GPTQ (Egiazarian et al., 2026) and ARCQuant (Meng et al., 2026) are evaluated under both MXFP4 and NVFP4, whereas SERQ (Park et al., 2026) is evaluated only under its officially supported MXFP4 setting. For training-based recovery, QAD (Xin et al., 2026) is the strongest directly matched full-parameter reference under its published NVFP4 setting and outperforms conventional QAT in the source study.

**Implementation Details.** We fix $K = 128$ across models, matching the correction width used by SERQ (Park et al., 2026) for a controlled comparison. LoDiQ-Lite uses channel-localized correction only, LoDiQ-Pro applies GRCA at 50% of the full residual-parameter budget, and LoDiQ-Max compensates every quantized linear layer. Channel localization and GRCA use C4 calibration data

Table 1: Comparison with state-of-the-art PTQ baselines under MXFP4 W4A4.

| Format | Method | Llama-3.2-3B | | | Llama-3-8B | | | Qwen3-8B | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | WT2 ↓ | 0-shot Avg. ↑ | MMLU ↑ | WT2 ↓ | 0-shot Avg. ↑ | MMLU ↑ | WT2 ↓ | 0-shot Avg. ↑ | MMLU ↑ |
| BF16 | baseline | 7.810 | 68.32 | 56.41 | 6.140 | 73.38 | 65.48 | 9.720 | 70.43 | 74.93 |
| MXFP4 | RTN | 9.920 | 62.40 | 46.87 | 7.960 | 68.85 | 56.27 | 10.910 | 65.93 | 69.51 |
| | MR-GPTQ | 11.110 | 63.15 | 46.76 | 7.680 | 68.79 | 56.59 | 10.730 | 66.07 | 69.22 |
| | SERQ | 10.940 | 62.01 | 48.25 | 7.670 | 68.68 | 56.37 | 11.040 | 67.69 | 70.87 |
| | ARCQuant | **9.270** | 63.56 | 48.82 | **7.520** | 69.92 | 57.66 | 10.770 | 66.61 | 70.82 |
| | LoDiQ-Lite | 10.120 | 64.22 | 49.91 | 7.970 | 70.50 | 58.45 | 9.880 | 69.03 | 71.43 |
| | LoDiQ-Pro | 9.620 | **65.47** | **50.50** | 7.690 | 70.83 | 59.40 | **9.730** | **69.94** | **72.16** |
| | LoDiQ-Max | 9.590 | 65.09 | 50.34 | 7.690 | **71.52** | **59.46** | 9.730 | 69.75 | 72.00 |

| Format | Method | Qwen3-32B | | | Moonlight-16B-A3B-Instruct | | |
|---|---|---|---|---|---|---|---|
| | | WT2 ↓ | 0-shot Avg. ↑ | MMLU ↑ | WT2 ↓ | 0-shot Avg. ↑ | MMLU ↑ |
| BF16 | baseline | 7.610 | 74.85 | 81.87 | 8.459 | 73.77 | 69.65 |
| MXFP4 | RTN | 8.920 | 71.99 | 79.06 | 11.021 | 68.40 | 61.60 |
| | MR-GPTQ | 8.370 | 72.35 | 79.63 | – | – | – |
| | SERQ | 8.850 | 70.74 | 79.52 | – | – | – |
| | ARCQuant | 8.740 | 72.34 | 79.70 | – | – | – |
| | LoDiQ-Lite | 7.580 | 74.32 | 80.10 | 9.594 | **70.66** | 63.69 |
| | LoDiQ-Pro | 7.510 | **74.79** | 80.47 | **9.393** | 70.52 | **64.53** |
| | LoDiQ-Max | **7.490** | 74.41 | **80.69** | 9.422 | 70.48 | 64.25 |

Table 2: Comparison with state-of-the-art PTQ baselines under NVFP4 W4A4.

| Format | Method | Llama-3.2-3B | | | Llama-3-8B | | | Qwen3-8B | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | WT2 ↓ | 0-shot Avg. ↑ | MMLU ↑ | WT2 ↓ | 0-shot Avg. ↑ | MMLU ↑ | WT2 ↓ | 0-shot Avg. ↑ | MMLU ↑ |
| BF16 | baseline | 7.810 | 68.32 | 56.41 | 6.140 | 73.38 | 65.48 | 9.720 | 70.43 | 74.93 |
| NVFP4 | RTN | 8.630 | 65.93 | 52.26 | 6.870 | 71.10 | 62.04 | 10.110 | 68.69 | 72.62 |
| | MR-GPTQ | 8.640 | 66.13 | 52.59 | 6.850 | 71.49 | 61.91 | 10.260 | 69.21 | 73.04 |
| | ARCQuant | **8.570** | 65.49 | 52.78 | **6.810** | 72.33 | 62.16 | 10.340 | 68.82 | 73.17 |
| | LoDiQ-Lite | 8.680 | **66.85** | 51.65 | 6.950 | 72.24 | 62.06 | 9.620 | 70.08 | 72.97 |
| | LoDiQ-Pro | 8.580 | 66.44 | 52.73 | 6.860 | 72.25 | 62.08 | 9.570 | 70.22 | **73.47** |
| | LoDiQ-Max | 8.580 | 66.43 | **52.82** | 6.850 | **72.36** | **62.58** | **9.560** | **70.35** | **73.47** |

| Format | Method | Qwen3-32B | | | Moonlight-16B-A3B-Instruct | | |
|---|---|---|---|---|---|---|---|
| | | WT2 ↓ | 0-shot Avg. ↑ | MMLU ↑ | WT2 ↓ | 0-shot Avg. ↑ | MMLU ↑ |
| BF16 | baseline | 7.610 | 74.85 | 81.87 | 8.459 | 73.77 | 69.65 |
| NVFP4 | RTN | 7.990 | 74.21 | 80.96 | 9.764 | 69.51 | 64.90 |
| | MR-GPTQ | 7.880 | 74.22 | 80.95 | – | – | – |
| | ARCQuant | 7.930 | 74.03 | 81.04 | – | – | – |
| | LoDiQ-Lite | 7.360 | 74.68 | **81.39** | 8.842 | 70.97 | 65.78 |
| | LoDiQ-Pro | 7.350 | 74.79 | 81.22 | **8.756** | 71.01 | 65.98 |
| | LoDiQ-Max | **7.340** | **74.90** | 81.30 | 8.825 | **71.73** | **66.04** |

(Raffel et al., 2020), and distillation uses 1M tokens sampled from AceReason-1.1-SFT (Liu et al., 2026). Distillation uses a $10^{-5}$ learning rate on NVIDIA H20 GPUs, where BF16 fake-quantized student tensors simulate the target FP4 formats. Inference is benchmarked on an NVIDIA RTX 5090. Appendix A.1 provides the detailed hyperparameters and evaluation protocols.

### 4.2 Main Results

**Accuracy comparison.** Tables 1 and 2 report W4A4 accuracy for the Llama and Qwen families. In both FP4 formats, LoDiQ surpasses matched PTQ baselines on downstream tasks, with higher residual-parameter budgets generally yielding stronger recovery. On Qwen3-8B, LoDiQ improves the zero-shot average over the strongest PTQ baseline by 2.25 points with MXFP4 and 1.14 points with NVFP4. On Qwen3-32B, the best LoDiQ result matches BF16 within 0.06 points for MXFP4 and NVFP4, indicating nearly lossless W4A4 quantization. Moonlight-16B-A3B-Instruct further confirms compatibility with an MoE architecture.

### 4.3 Training and Inference Efficiency

**Training cost.** Table 3 reports measured LoDiQ training cost on Qwen3-32B. Across residual-parameter budgets, LoDiQ updates 587M–1.174B parameters, corresponding to only 1.79–3.59% of the model, and completes one epoch over 1M supervised tokens in 4.49–4.67 GPU-hours on a single H20. The full-QAD row is an analytical estimate rather than a measured run. Following the FP32 master-weight setup in the source QAD study (Xin et al., 2026), the estimate counts FP32 parameters, gradients, and two AdamW moment tensors, totaling at least 488.19 GB before

Table 3: Training cost on Qwen3-32B using one NVIDIA H20 GPU.

| Method | Trainable params | Trainable ratio[‡] | FP32 params (GB) | FP32 trainable state[§] (GB) | Peak allocated (GB) | Optimizer steps | Avg. s/step | Training GPU-hours | Teacher cache size (GB)[*] |
|---|---|---|---|---|---|---|---|---|---|
| Full QAD[†] | 32.76B | 100% | 122.05 | 488.19 | $\geq$488.19 | – | – | – | 304.48 |
| LoDiQ-Lite | 587.20M | 1.79% | 2.188 | 8.750 | 77.01 | 679 | 23.78 | 4.49 | 0.098 |
| LoDiQ-Pro | 880.80M | 2.69% | 3.281 | 13.125 | 81.32 | 679 | 24.33 | 4.59 | 0.098 |
| LoDiQ-Max | 1.174B | 3.59% | 4.375 | 17.500 | 85.69 | 679 | 24.77 | 4.67 | 0.098 |

[†]Analytical FP32 training-state estimate; not a measured run. [‡]Trainable parameters divided by the 32.76B model parameters. [§]FP32 parameters, gradients, and two AdamW moment tensors. [*]Compact teacher cache for 1M supervised tokens, storing teacher top-16 token targets and aggregate vocabulary-remainder mass.

Table 4: End-to-end GPU performance on Qwen3-8B using one NVIDIA RTX 5090. Ratios are relative to the paired BF16 configuration.

| Format | Method | Batch 8 TTFT (ms) ↓ | Batch 8 TPOT (ms) ↓ | Batch 8 Peak mem. (GiB) ↓ | Batch 16 TTFT (ms) ↓ | Batch 16 TPOT (ms) ↓ | Batch 16 Peak mem. (GiB) ↓ |
|---|---|---|---|---|---|---|---|
| **MXFP4** | | | | | | | |
| BF16 | baseline | 926.5 (1.00×) | 14.14 (1.00×) | 21.87 (1.00×) | 2072.2 (1.00×) | 15.87 (1.00×) | 27.28 (1.00×) |
| MXFP4 | RTN | 271.0 (3.42×) | 9.96 (1.42×) | 12.61 (1.73×) | 627.1 (3.30×) | 11.69 (1.36×) | 18.04 (1.51×) |
| | LoDiQ-Lite | 308.4 (3.00×) | 10.08 (1.40×) | 12.61 (1.73×) | 647.3 (3.20×) | 11.94 (1.33×) | 18.04 (1.51×) |
| | LoDiQ-Pro | 315.7 (2.93×) | 10.39 (1.36×) | 12.77 (1.71×) | 664.0 (3.12×) | 12.21 (1.30×) | 19.73 (1.38×) |
| | LoDiQ-Max | 315.8 (2.93×) | 10.40 (1.36×) | 12.89 (1.70×) | 666.1 (3.11×) | 12.23 (1.30×) | 18.32 (1.49×) |
| **NVFP4** | | | | | | | |
| BF16 | baseline | 1003.9 (1.00×) | 13.94 (1.00×) | 21.88 (1.00×) | 2045.0 (1.00×) | 15.85 (1.00×) | 27.28 (1.00×) |
| NVFP4 | RTN | 324.0 (3.10×) | 10.77 (1.29×) | 12.82 (1.71×) | 653.2 (3.13×) | 12.74 (1.24×) | 18.25 (1.49×) |
| | LoDiQ-Lite | 319.4 (3.14×) | 10.97 (1.27×) | 12.83 (1.71×) | 696.1 (2.94×) | 12.81 (1.24×) | 18.25 (1.49×) |
| | LoDiQ-Pro | 346.4 (2.90×) | 11.83 (1.18×) | 13.10 (1.67×) | 716.1 (2.86×) | 13.66 (1.16×) | 20.01 (1.36×) |
| | LoDiQ-Max | 334.0 (3.01×) | 12.26 (1.14×) | 13.94 (1.57×) | 745.4 (2.74×) | 13.98 (1.13×) | 19.37 (1.41×) |

Table 5: Component ablation on Qwen3-8B under MXFP4 W4A4.

| Learning signal | Method | Error-localized reordering | Residual compensation | GRCA | Training | Budget | WT2 ↓ | 0-shot Avg. ↑ | MMLU ↑ |
|---|---|---|---|---|---|---|---|---|---|
| None | RTN | × | × | × | × | – | 10.910 | 65.93 | 69.51 |
| | LoDiQ-Lite | ✓ | × | × | × | 0% | 11.079 | 66.26 | 69.14 |
| | LoDiQ-Pro | ✓ | ✓ | ✓ | × | 50% | 10.702 | 67.32 | 70.40 |
| | LoDiQ-Max | ✓ | ✓ | × | × | 100% | 10.675 | 67.19 | 70.74 |
| Teacher KL | LoDiQ-Lite | ✓ | × | × | ✓ | 0% | 9.880 | 69.03 | 71.43 |
| | LoDiQ-Pro | ✓ | ✓ | ✓ | ✓ | 50% | **9.730** | **69.94** | **72.16** |
| | LoDiQ-Max | ✓ | ✓ | × | ✓ | 100% | **9.730** | 69.75 | 72.00 |

activations and runtime buffers. LoDiQ's measured peak allocation, including runtime allocations, remains within 17.6% of this conservative lower bound. As a secondary storage saving, compact teacher targets reduce the 1M-token cache from 304.48 to 0.098 GB (3,107×), while preserving KL correlations above 0.996 and a mean gradient cosine similarity of 0.9616 on held-out tokens (Appendix Figure 3).

**End-to-end GPU performance.** Table 4 reports time to first token (TTFT), time per output token (TPOT), and peak memory on Qwen3-8B using 2,048-token prompts and 512-token generations. Across the three residual budgets and batch sizes 8 and 16, LoDiQ achieves 2.93–3.20× TTFT speedups over BF16 with MXFP4 and 2.74–3.14× with NVFP4. Comparison with format-matched RTN quantifies the incremental cost of the expanded low-bit GEMM. At batch size 16, the Lite-to-Max TTFT overhead ranges from 3.2 to 6.2% under MXFP4 and from 6.6 to 14.1% under NVFP4. Increasing the residual budget from Lite to Max raises TTFT by only 2.4–2.9% and 4.6–7.1% across the two batch sizes, respectively. The packed expansion therefore retains most of the native FP4 prefill efficiency across all budgets. LoDiQ further provides 1.30–1.40× and 1.13–1.27× TPOT speedups, while reducing peak memory by 27.7–42.3% and 26.6–41.4% under MXFP4 and NVFP4, respectively. Complete throughput results are reported in Appendix Table 13.

### 4.4 Ablation Studies

Table 5 separates the effects of error-localized reordering, residual compensation, and model-level distillation. Untrained LoDiQ variants retain the initialization and residual allocation of their trained counterparts but receive no optimizer update.

Table 6: Reasoning accuracy (%) on Qwen3-8B under MXFP4 and NVFP4.

| Format | Method | GPQA-D Avg@4 ↑ | GSM8K 5-shot ↑ |
|---|---|---|---|
| BF16 | baseline | **59.60** | 93.63 |
| MXFP4 | RTN | 47.47 | 89.99 |
| | LoDiQ-Max | **52.27** | **92.95** |
| NVFP4 | RTN | 53.03 | 91.28 |
| | LoDiQ-Max | **56.69** | **93.25** |

Table 7: NVFP4 reasoning accuracy (%) on AceReason Nemotron 1.1 7B.

| Method | AIME25 Avg@30 ↑ | AIME26 Avg@30 ↑ |
|---|---|---|
| BF16 | 65.22 | 69.22 |
| RTN | 59.78 | 64.44 |
| QAD 1M | 59.89 | 66.44 |
| QAD 100M | 61.89 | 66.00 |
| LoDiQ-Max | **62.33** | **67.67** |

**Recovery components.** Without optimization, error-localized reordering alone changes the zero-shot average only modestly, from 65.93 for RTN to 66.26 for LoDiQ-Lite. Adding the residual representation increases the untrained score to 67.32 for LoDiQ-Pro and 67.19 for LoDiQ-Max, showing that the representation itself recovers part of the quantization loss. Model-level distillation provides the larger gain: it adds 2.77, 2.62, and 2.56 zero-shot points over the untrained Lite, Pro, and Max variants, respectively, while reducing perplexity for all three. The two effects are complementary rather than interchangeable.

**Residual-budget allocation.** With teacher KL, LoDiQ-Pro matches LoDiQ-Max at 9.73 perplexity and improves the zero-shot average and MMLU by 0.19 and 0.16 points, despite using half of the residual-parameter budget. The result supports GRCA as a cost-aware allocation rule and also explains why a larger residual budget need not produce monotonic downstream gains.

**Additional reasoning tasks.** Table 6 reports generated-answer reasoning accuracy on Qwen3-8B. LoDiQ-Max improves GPQA-Diamond over format-matched RTN by 4.80 points under MXFP4 and 3.66 points under NVFP4. On five-shot GSM8K, LoDiQ-Max reaches 92.95 and 93.25, retaining 99.27% and 99.59% of BF16 accuracy, respectively. On the matched QAD reproduction in Table 7, LoDiQ-Max outperforms the 1M- and 100M-token QAD runs on both AIME25 and AIME26. Relative to the 100M-token run, LoDiQ uses 1/100 as many distillation tokens and 4.67% as many trainable parameters (356.0M versus 7.616B).

**Data and token sensitivity.** Appendix Tables 14 and 15 examine two additional sources of variation. Meaningful sequences consistently outperform random tokens in perplexity and decode KL, although AceReason, RedPajama, and teacher-generated sequences trade places on downstream accuracy. Increasing the budget from 1M to 10M tokens improves MXFP4 zero-shot accuracy by 0.59 points and reduces perplexity in both formats, but NVFP4 downstream accuracy is already strongest at 1M tokens. The default therefore prioritizes low training cost rather than claiming that every metric has saturated.

## 5 Conclusion

In this work, we introduced LoDiQ, an error-localized distillation framework designed to bridge efficient PTQ and costly full-model QAD. Unlike PTQ based on local calibration and QAD based on model-wide optimization, LoDiQ preserves model-level teacher supervision while concentrating trainable recovery capacity on quantization-sensitive parameters. Propagated-error channel localization determines where updates are needed, quantized residual compensation strengthens difficult linear operations, and GRCA allocates this capacity under a global budget. Compact teacher targets further reduce training storage. Together, these components improve accuracy over strong PTQ baselines while substantially reducing the trainable state and distillation data required by full-model QAD. The learned correction remains in the target low-bit representation and is packed with frozen weights into expanded GEMMs, retaining a uniform and hardware-friendly inference path. More broadly, the "distill globally, train locally" principle is not restricted to FP4; it provides a general framework for localized recovery across numerical formats and quantization bit-width budgets.

### AI USE STATEMENT

Generative AI tools were used to support research ideation and execution, including methodological and experimental development, and to assist with manuscript drafting and language polishing. All AI-assisted work was reviewed by the authors, who take responsibility for the final content.

### REPRODUCIBILITY STATEMENT

Section 4 specifies the quantization boundary, datasets, evaluation tasks, hardware, and measurement protocol. The method and all optimization objectives are defined in Sections 3–3.3. Appendix Section A.1 lists the calibration, allocation, and distillation hyperparameters, and Appendix Section A.3 provides per-task and runtime results. Together, these sections document the information required to reproduce the reported operating points and comparisons. An anonymized implementation, experiment manifest, and reproduction instructions are available at `https://anonymous.4open.science/r/lodiq-D4B7A91C`. The same source snapshot is included with the supplementary material.

## REFERENCES

Eduardo Alvarez, Omri Almog, Eric Chung, Simon Layton, Dusan Stosic, Ronny Krashinsky, and Kyle Aubrey. Introducing NVFP4 for efficient and accurate low-precision inference. NVIDIA Technical Blog, June 2025. URL https://developer.nvidia.com/blog/introducing-nvfp4-for-efficient-and-accurate-low-precision-inference/.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-free 4-bit inference in rotated LLMs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 100213–100240. Curran Associates, Inc., 2024. doi: 10.52202/079017-3180. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/b5b939436789f76f08b9d0da5e81af7c-Paper-Conference.pdf.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239.

Zhixuan Chen, Xing Hu, Dawei Yang, Zukang Xu, Xu Chen, Zhihang Yuan, Sifan Zhou, and Jiangyong Yu. MoEQuant: Enhancing quantization for mixture-of-experts large language models via expert-balanced sampling and affinity guidance. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 8245–8260. PMLR, 2025. URL https://proceedings.mlr.press/v267/chen25aa.html.

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. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115. Curran Associates, Inc., 2023. doi: 10.52202/075280-0441. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1feb87871436031bdc0f2beaa62a049b-Paper-Conference.pdf.

Xin Ding, Xiaoyu Liu, Zhijun Tu, Yun Zhang, Wei Li, Jie Hu, Hanting Chen, Yehui Tang, Zhiwei Xiong, Baoqun Yin, and Yunhe Wang. CBQ: Cross-block quantization for large language models. In *International Conference on Learning Representations*, pp. 7056–7075, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/15212bd2265c4a3ab0dbc1b1982c1b69-Paper-Conference.pdf.

Vage Egiazarian, Roberto L. Castro, Denis Kuznedelev, Andrei Panferov, Eldar Kurtic, Shubhra Pandit, Alexandre Marques, Mark Kurtz, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Bridging the gap between promise and performance for microscaling FP4 quantization. In *International Conference on Learning Representations*, pp. 113529–113563, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/b87bb4f6346d727b265088235e5bc389-Paper-Conference.pdf.

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.

Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2009.03300.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. DuQuant: Distributing outliers via dual transformation makes stronger quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 87766–87800. Curran Associates, Inc., 2024. doi: 10.52202/079017-2786. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9febda1c8344cc5f2d51713964864e93-Paper-Conference.pdf.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is scalable for LLM training, 2025a. URL https://arxiv.org/abs/2502.16982.

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*, pp. 92009–92032, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/e5b1c0d4866f72393c522c8a00eed4eb-Paper-Conference.pdf.

Zihan Liu, Zhuolin Yang, Yang Chen, Chankyu Lee, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. AceReason-Nemotron 1.1: Advancing math and code reasoning through SFT and RL synergy. In *International Conference on Learning Representations*, pp. 70010–70028, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/72398f80dc0f81159f79e6ff8fd1e376-Paper-Conference.pdf.

Haoqian Meng, Yilun Luo, Yafei Zhao, Wenyuan Liu, Peng Zhang, and Xindian Ma. ARCQuant: Boosting NVFP4 quantization with augmented residual channels for LLMs. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8609–8623, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.388. URL https://aclanthology.org/2026.acl-long.388/.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1609.07843.

Meta. Llama 3.2 model card. Official model card, 2024. URL https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md.

Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021.

Open Compute Project. OCP microscaling formats (MX) specification. Technical Report Version 1.0, Open Compute Project Foundation, September 2023. URL https://www.opencompute.org/documents/ocp-microscaling-formats-mx-v1-0-spec-final-pdf.

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 *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1525–1534, 2016. doi: 10.18653/v1/P16-1144. URL https://aclanthology.org/P16-1144/.

Yeonsik Park, Hyeonseong Kim, and Seungkyu Choi. SERQ: Saliency-aware low-rank error reconstruction for LLM quantization. In *International Conference on Learning Representations*, pp. 35889–35909, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/3ca380c5fe9f174a71a230478741169f-Paper-Conference.pdf.

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. URL https://www.jmlr.org/papers/v21/20-074.html.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-Proof Q&A benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Bita Darvish Rouhani, Ritchie Zhao, Ankit More, Mathew Hall, Alireza Khodamoradi, Summer Deng, Dhruv Choudhary, Marius Cornea, Eric Dellinger, Kristof Denolf, Dusan Stosic, Venmugil Elango, Maximilian Golub, Alexander Heinecke, Phil James-Roxby, Dharmesh Jani, Gaurav Kolhe, Martin Langhammer, Ada Li, Levi Melnick, Maral Mesmakhosroshahi, Andres Rodriguez, Michael Schulte, Rasoul Shafipour, Lei Shao, Michael Siu, Pradeep Dubey, Paulius Micikevicius, Maxim Naumov, Colin Verrilli, Ralph Wittig, Doug Burger, and Eric Chung. Microscaling data formats for deep learning. *arXiv preprint arXiv:2310.10537*, 2023. URL https://arxiv.org/abs/2310.10537.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020. doi: 10.1609/aaai.v34i05.6399.

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.

Meng Xin, Sweta Priyadarshi, Jingyu Xin, Bilal Kartal, Aditya Vavre, Asma Kuriparambil Thekkumpate, Zijia Chen, Ameya Sunil Mahabaleshwarkar, Ido Shahaf, Akhiad Bercovich, Kinjal Patel, Suguna Varshini Velury, Chenjie Luo, Zhiyu Cheng, Jenny Chen, Chen-Han Yu, Wei Ping, Oleg Rybakov, Nima Tajbakhsh, Oluwatobi Olabiyi, Dusan Stosic, Di Wu, Song Han, Eric Chung, Sharath Turuvekere Sreenivas, Bryan Catanzaro, Yoshi Suhara, Tijmen Blankevoort, and Huizi Mao. Quantization-aware distillation for NVFP4 inference accuracy recovery. *arXiv preprint arXiv:2601.20088*, 2026. URL https://arxiv.org/abs/2601.20088.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang,

Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Yifan Zhang and Math-AI Team. American Invitational Mathematics Examination (AIME) 2025, 2025. URL https://huggingface.co/datasets/math-ai/aime25.

Yifan Zhang and Math-AI Team. American Invitational Mathematics Examination (AIME) 2026, 2026. URL https://huggingface.co/datasets/math-ai/aime26.

# A Appendix

## A.1 Implementation and Training Details

Table 8: Default calibration and distillation hyperparameters.

| Setting | Value | Setting | Value |
|---|---|---|---|
| Channel calibration data | C4 train shard | Calibration sequences | 128 |
| Calibration sequence length | 2,048 | Calibration seed | 42 |
| GRCA scoring data | C4 train shard | GRCA scoring sequences | 64 |
| GRCA sequence / scored tokens | 512 / 511 | GRCA confidence coefficient | 1.0 |
| Selected-channel width $K$ | 128 | Quantized group size | 32 (MXFP4) / 16 (NVFP4) |
| Distillation data | AceReason-1.1-SFT | Maximum sequence length | 2,048 |
| Training / validation tokens | 1.002M / 0.101M | Compact target | top-16 plus remainder |
| Teacher precision | BF16 | Trainable corrective weights | FP32 master |
| Frozen model parameters | FP4 (BF16 fake QDQ on H20) | Student forward | target-format fake QDQ |
| Optimizer | AdamW | Learning rate | $10^{-5}$ |
| Weight decay / warmup | 0 / none | Gradient-norm limit | 1.0 |
| Epochs | 1 | Micro-batch / accumulation | 1 / 1 |

**NVFP4 runtime calibration and memory accounting.** For the runtime measurements, each frozen packed student is calibrated on the same 128 C4 training sequences of length 2,048 with seed 42. Input quantization sites are calibrated in network order, with upstream scales fixed before collecting downstream activation maxima. At each site, the FP32 global scale is fixed to $\max(a_{\max}/2688, 10^{-8})$; block-16 scales remain dynamic. This calibration changes neither the learned weights nor the retained QKV/gate-up fusion, and its cost is excluded from steady-state inference timing. Peak memory in the runtime tables is serving-phase NVML usage, including allocator-reserved memory, rather than PyTorch allocated memory alone. For NVFP4 LoDiQ-Pro at batch 16, the static-scale run reaches 20.01 GiB NVML usage; its PyTorch allocated peak is 15.502 GiB, effectively unchanged from the paired online-scale run, while the reserved peak increases from 16.744 to 18.740 GiB. The table retains the measured NVML peak.

## A.2 GRCA Scoring and Budget Optimization

For Eq. 7, the output, gradient, and error tensors are vectorized over token positions and output channels. The squared norms sum over both dimensions, and each calibration sequence contributes one $\Delta_i$. The NLL is averaged over the sequence's scored next-token positions before its gradient is computed. The default setup uses $N = 64$ sequences of length 512, with 511 scored positions per sequence. The sample standard deviation in Eq. 7 is computed across sequences.

Equation 8 is a 0–1 knapsack problem with layer scores $G_l$ and integer parameter costs $c_l$. Costs are divided by their greatest common divisor and the budget is rounded down in the resulting units before exact dynamic programming. All primary weight blocks $\boldsymbol{W}_T$ remain trainable regardless of the allocation. The full-budget endpoint directly enables residual compensation in every quantized linear layer rather than applying the score-based selection. Scoring performs no parameter updates, and teacher KL is used for the subsequent distillation.

## A.3 Additional Results

**Compact-target fidelity.** Figure 3 compares the compact objective with full-vocabulary KL on held-out tokens. The two token-level losses have Pearson and Spearman correlations of 0.9986 and 0.9969, respectively. Their gradients also attain a mean cosine similarity of 0.9616, showing that the compact targets closely preserve the supervision signal in this configuration.

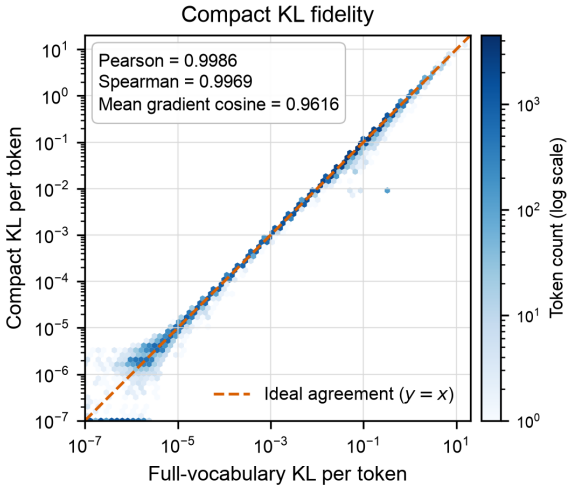


Figure 3: Compact-target fidelity on Qwen3-8B. Token-level compact KL is compared with full-vocabulary KL for the MXFP4 LoDiQ-Max student on 128 held-out sequences (131,072 tokens). The dashed diagonal denotes exact agreement, and hexagon intensity represents token density.

**Per-task accuracy.** Tables 9 and 10 report the six tasks underlying the aggregate results in Tables 1 and 2. The best LoDiQ operating point improves the zero-shot average over the strongest matched PTQ baseline for every model, by 1.60–2.44 points under MXFP4 and 0.03–1.14 points under NVFP4. On Qwen3-32B, the resulting averages of 74.79 and 74.90 are within 0.06 points of the BF16 reference in both formats.

**Extension to INT4.** Table 11 evaluates INT4 quantization on Qwen3-8B as a format extension beyond MXFP4 and NVFP4. LoDiQ-Max corresponds to the full residual-compensation budget. Compared with ARCQuant, LoDiQ-Max improves the zero-shot average by 7.02 points and MMLU by 5.20 points, while reducing WikiText-2 perplexity from 12.31 to 10.01. This single-model extension suggests that error-localized distillation is not specific to microscaling floating-point formats.

**MoE transfer.** Table 12 provides the complete results on Moonlight-16B-A3B-Instruct. Every LoDiQ operating point improves RTN in WikiText-2 perplexity, zero-shot average, and MMLU under both formats. The best zero-shot averages increase from 68.40 to 70.66 under MXFP4 and from 69.51 to 71.73 under NVFP4, supporting the applicability of channel-localized recovery to an MoE architecture without an expert-specific design.

**Complete runtime results.** Table 13 expands the main runtime comparison to both batch sizes and all three LoDiQ budgets. Across these settings, LoDiQ retains 2.74–3.20× TTFT speedups, 1.23–1.51× higher output throughput, and 1.36–1.73× lower peak memory than BF16. Relative to the format-matched RTN kernels, the expanded low-bit operations preserve most of the available throughput while providing the accuracy recovery reported in the main results.

Table 9: Per-task MXFP4 W4A4 results corresponding to Table 1.

| Format | Method | Zero-shot accuracy ↑ Arc-C | Hella. | LAMB. | PIQA | Wino | Arc-E | 0-shot Avg. | PPL ↓ WikiText-2 | MMLU ↑ 5-shot |
|---|---|---|---|---|---|---|---|---|---|---|
| **Llama-3.2-3B** | | | | | | | | | | |
| BF16 | baseline | 46.33 | 74.11 | 69.67 | 78.07 | 69.69 | 72.05 | 68.32 | 7.81 | 56.41 |
| MXFP4 | RTN | 40.44 | 69.16 | 62.08 | 73.56 | 64.17 | 64.98 | 62.40 | 9.92 | 46.87 |
| | MR-GPTQ | 41.13 | 68.55 | 63.32 | 74.16 | 65.19 | 66.54 | 63.15 | 11.11 | 46.76 |
| | SERQ | 40.53 | 67.78 | 62.33 | 73.99 | 64.48 | 62.92 | 62.01 | 10.94 | 48.25 |
| | ARCQuant | 41.81 | 69.16 | 64.84 | 75.24 | 64.25 | 66.04 | 63.56 | **9.27** | 48.82 |
| | LoDiQ-Lite | 42.32 | 71.60 | 66.62 | 75.14 | 64.25 | 65.40 | 64.22 | 10.12 | 49.91 |
| | LoDiQ-Pro | **43.69** | 71.74 | 68.85 | **76.01** | **65.59** | **66.96** | **65.47** | 9.62 | **50.50** |
| | LoDiQ-Max | **43.69** | **71.87** | **69.03** | 75.14 | 64.40 | 66.41 | 65.09 | 9.59 | 50.34 |
| **Llama-3-8B** | | | | | | | | | | |
| BF16 | baseline | 53.92 | 79.29 | 75.55 | 80.52 | 73.48 | 77.53 | 73.38 | 6.14 | 65.48 |
| MXFP4 | RTN | 47.87 | 74.59 | 69.84 | 77.37 | 71.19 | 72.22 | 68.85 | 7.96 | 56.27 |
| | MR-GPTQ | 46.50 | 75.37 | 69.30 | 77.64 | 69.46 | 74.49 | 68.79 | 7.68 | 56.59 |
| | SERQ | 47.70 | 74.74 | 71.90 | 76.71 | 68.59 | 72.43 | 68.68 | 7.67 | 56.37 |
| | ARCQuant | 49.23 | 75.39 | 72.09 | 78.40 | 69.85 | 74.58 | 69.92 | **7.52** | 57.66 |
| | LoDiQ-Lite | 48.46 | 76.81 | 72.02 | 78.56 | 71.98 | **75.17** | 70.50 | 7.97 | 58.45 |
| | LoDiQ-Pro | 50.09 | 77.30 | 72.21 | **79.54** | 71.19 | 74.66 | 70.83 | 7.69 | 59.40 |
| | LoDiQ-Max | **52.13** | **77.40** | **72.68** | 79.11 | **72.69** | 75.08 | **71.52** | 7.69 | **59.46** |
| **Qwen3-8B** | | | | | | | | | | |
| BF16 | baseline | 56.74 | 74.90 | 64.12 | 77.80 | 68.19 | 80.85 | 70.43 | 9.72 | 74.93 |
| MXFP4 | RTN | 50.68 | 70.44 | 59.15 | 75.41 | 66.14 | 73.78 | 65.93 | 10.91 | 69.51 |
| | MR-GPTQ | 50.26 | 70.60 | 57.91 | 75.41 | 66.06 | 76.18 | 66.07 | 10.73 | 69.22 |
| | SERQ | 52.30 | 71.60 | 63.90 | 74.21 | 68.59 | 75.51 | 67.69 | 11.04 | 70.87 |
| | ARCQuant | 52.47 | 71.27 | 62.27 | 74.10 | 64.48 | 75.04 | 66.61 | 10.77 | 70.82 |
| | LoDiQ-Lite | **57.25** | 73.43 | 60.68 | 76.66 | 67.01 | **79.12** | 69.03 | 9.88 | 71.43 |
| | LoDiQ-Pro | 56.31 | **73.62** | **63.94** | **77.31** | **69.85** | 78.62 | **69.94** | **9.73** | **72.16** |
| | LoDiQ-Max | 56.66 | 73.58 | 63.01 | 77.26 | 69.14 | 78.83 | 69.75 | 9.73 | 72.00 |
| **Qwen3-32B** | | | | | | | | | | |
| BF16 | baseline | 60.84 | 82.59 | 67.13 | 81.88 | 73.40 | 83.25 | 74.85 | 7.61 | 81.87 |
| MXFP4 | RTN | 60.15 | 79.79 | 62.95 | 79.54 | 68.59 | 80.93 | 71.99 | 8.92 | 79.06 |
| | MR-GPTQ | 58.87 | 80.69 | 63.65 | 78.56 | 70.96 | 81.36 | 72.35 | 8.37 | 79.63 |
| | SERQ | 57.08 | 79.82 | 61.44 | 78.62 | 68.75 | 78.70 | 70.74 | 8.85 | 79.52 |
| | ARCQuant | 60.15 | 80.07 | 63.23 | 79.38 | 70.09 | 81.14 | 72.34 | 8.74 | 79.70 |
| | LoDiQ-Lite | 61.09 | 81.62 | 64.35 | **81.12** | 75.06 | 82.70 | 74.32 | 7.58 | 80.10 |
| | LoDiQ-Pro | **62.29** | **81.79** | **66.14** | 80.69 | **75.30** | 82.53 | **74.79** | 7.51 | 80.47 |
| | LoDiQ-Max | 60.75 | 81.58 | 65.59 | 81.07 | 74.03 | **83.42** | 74.41 | **7.49** | **80.69** |

Table 10: Per-task NVFP4 W4A4 results corresponding to Table 2.

| Format | Method | Zero-shot accuracy ↑ Arc-C | Hella. | LAMB. | PIQA | Wino | Arc-E | 0-shot Avg. | PPL ↓ WikiText-2 | MMLU ↑ 5-shot |
|---|---|---|---|---|---|---|---|---|---|---|
| **Llama-3.2-3B** | | | | | | | | | | |
| BF16 | baseline | 46.33 | 74.11 | 69.67 | 78.07 | 69.69 | 72.05 | 68.32 | 7.81 | 56.41 |
| NVFP4 | RTN | 43.52 | 71.46 | 66.82 | 76.06 | 67.25 | **70.50** | 65.93 | 8.63 | 52.26 |
| | MR-GPTQ | 43.34 | 71.97 | 67.34 | 76.93 | 67.56 | 69.65 | 66.13 | 8.64 | 52.59 |
| | ARCQuant | 44.62 | 71.14 | 67.81 | 75.24 | 66.14 | 68.01 | 65.49 | **8.57** | 52.78 |
| | LoDiQ-Lite | **45.65** | 72.63 | 68.04 | 76.01 | **69.85** | 68.94 | 66.85 | 8.68 | 51.65 |
| | LoDiQ-Pro | 45.39 | **72.72** | 67.30 | **77.20** | 67.40 | 68.60 | **66.44** | 8.58 | 52.73 |
| | LoDiQ-Max | 43.52 | 72.69 | **68.39** | 76.12 | 68.03 | 69.82 | 66.43 | 8.58 | **52.82** |
| **Llama-3-8B** | | | | | | | | | | |
| BF16 | baseline | 53.92 | 79.29 | 75.55 | 80.52 | 73.48 | 77.53 | 73.38 | 6.14 | 65.48 |
| NVFP4 | RTN | 50.17 | 77.34 | 74.36 | 78.18 | 72.53 | 74.03 | 71.10 | 6.87 | 62.04 |
| | MR-GPTQ | 51.19 | 76.92 | 73.80 | 79.54 | 71.11 | 76.35 | 71.49 | 6.85 | 61.91 |
| | ARCQuant | 51.45 | 76.77 | **74.97** | 79.82 | **73.72** | **77.27** | 72.33 | **6.81** | 62.16 |
| | LoDiQ-Lite | **54.10** | **78.30** | 73.88 | 78.35 | 72.06 | 76.77 | 72.24 | 6.95 | 62.06 |
| | LoDiQ-Pro | 53.84 | 77.96 | 73.74 | 78.78 | 72.77 | 76.43 | 72.25 | 6.86 | 62.08 |
| | LoDiQ-Max | 52.39 | 78.10 | 73.70 | **79.71** | 73.09 | 77.15 | **72.36** | 6.85 | **62.58** |
| **Qwen3-8B** | | | | | | | | | | |
| BF16 | baseline | 56.74 | 74.90 | 64.12 | 77.80 | 68.19 | 80.85 | 70.43 | 9.72 | 74.93 |
| NVFP4 | RTN | 53.67 | 72.85 | 62.37 | 76.61 | 67.01 | 79.63 | 68.69 | 10.11 | 72.62 |
| | MR-GPTQ | 55.03 | 73.41 | 64.29 | 76.44 | 66.22 | 79.88 | 69.21 | 10.26 | 73.04 |
| | ARCQuant | 53.50 | 73.06 | 62.24 | 76.82 | 68.59 | 78.70 | 68.82 | 10.34 | 73.17 |
| | LoDiQ-Lite | 57.08 | 74.32 | 63.46 | 77.42 | 68.67 | 79.55 | 70.08 | 9.62 | 72.97 |
| | LoDiQ-Pro | 56.48 | 73.99 | 63.89 | **77.58** | 69.22 | **80.13** | 70.22 | 9.57 | **73.47** |
| | LoDiQ-Max | **57.17** | **74.37** | **64.33** | 77.20 | **69.38** | 79.63 | **70.35** | **9.56** | **73.47** |
| **Qwen3-32B** | | | | | | | | | | |
| BF16 | baseline | 60.84 | 82.59 | 67.13 | 81.88 | 73.40 | 83.25 | 74.85 | 7.61 | 81.87 |
| NVFP4 | RTN | 61.01 | 81.98 | 66.62 | 80.63 | 72.14 | 82.87 | 74.21 | 7.99 | 80.96 |
| | MR-GPTQ | 58.70 | 81.80 | 68.08 | 81.45 | 72.38 | 82.91 | 74.22 | 7.88 | 80.95 |
| | ARCQuant | 60.75 | 81.50 | 66.19 | 80.74 | 72.14 | 82.83 | 74.03 | 7.93 | 81.04 |
| | LoDiQ-Lite | 60.75 | 81.71 | 67.77 | 80.69 | **74.27** | 82.87 | 74.68 | 7.36 | **81.39** |
| | LoDiQ-Pro | **61.35** | 81.88 | 67.82 | **80.79** | 73.95 | 82.91 | 74.79 | 7.35 | 81.22 |
| | LoDiQ-Max | 60.92 | **82.10** | **67.98** | 80.74 | 74.03 | **83.63** | **74.90** | **7.34** | 81.30 |

Table 11: INT4 format extension on Qwen3-8B. Accuracy is reported in percent. Metrics and column ordering follow the detailed MXFP4 and NVFP4 results.

| Format | Method | Zero-shot accuracy ↑ Arc-C | Hella. | LAMB. | PIQA | Wino | Arc-E | 0-shot Avg. | PPL ↓ WikiText-2 | MMLU ↑ 5-shot |
|---|---|---|---|---|---|---|---|---|---|---|
| **Qwen3-8B** | | | | | | | | | | |
| BF16 | baseline | 56.74 | 74.90 | 64.12 | 77.80 | 68.19 | 80.85 | 70.43 | 9.7230 | 74.93 |
| INT4 | RTN | 43.60 | 63.71 | 48.26 | 72.47 | 58.64 | 67.80 | 59.08 | 12.8709 | 59.92 |
| | ARCQuant | 46.33 | 69.37 | 50.42 | 72.96 | 64.64 | 69.95 | 62.28 | 12.3109 | 65.65 |
| | LoDiQ-Max | **56.14** | **73.37** | **60.78** | **77.09** | **68.59** | **79.84** | **69.30** | **10.0119** | **70.85** |

Table 12: Detailed MoE transfer results on Moonlight-16B-A3B-Instruct. 0-shot Avg. is the unweighted mean over the six zero-shot tasks. No matched external baseline is available for this model.

| Format | Method | Zero-shot accuracy ↑ Arc-C | Hella. | LAMB. | PIQA | Wino | Arc-E | 0-shot Avg. | PPL ↓ WikiText-2 | MMLU ↑ 5-shot |
|---|---|---|---|---|---|---|---|---|---|---|
| **Moonlight-16B-A3B-Instruct** | | | | | | | | | | |
| BF16 | baseline | 58.36 | 80.40 | 70.95 | 80.74 | 70.40 | 81.78 | 73.77 | 8.459 | 69.65 |
| MXFP4 | RTN | 52.56 | 75.84 | 61.89 | 77.75 | 65.82 | 76.56 | 68.40 | 11.021 | 61.60 |
| | LoDiQ-Lite | **53.33** | 77.28 | **68.95** | 78.02 | **68.59** | 77.78 | **70.66** | 9.594 | 63.69 |
| | LoDiQ-Pro | **53.33** | **77.45** | 68.12 | **79.00** | 67.09 | **78.16** | 70.52 | **9.393** | **64.53** |
| | LoDiQ-Max | 52.73 | 77.37 | 68.23 | 78.73 | 68.27 | 77.53 | 70.48 | 9.422 | 64.25 |
| NVFP4 | RTN | 52.05 | 77.40 | 64.16 | 78.40 | 67.01 | **78.03** | 69.51 | 9.764 | 64.90 |
| | LoDiQ-Lite | 52.82 | 78.54 | 69.01 | 78.94 | 68.90 | 77.61 | 70.97 | 8.842 | 65.78 |
| | LoDiQ-Pro | 53.33 | **78.75** | **69.28** | 78.89 | 68.11 | 77.69 | 71.01 | **8.756** | 65.98 |
| | LoDiQ-Max | **54.61** | 78.72 | 69.09 | **79.33** | **70.72** | 77.90 | **71.73** | 8.825 | **66.04** |

Table 13: Complete Qwen3-8B inference results on one NVIDIA RTX 5090. Each configuration uses 2,048 input tokens, 512 output tokens, and three repetitions. Ratios use the paired BF16 reference within the same batch-size and format block. NVFP4 uses fixed, offline-calibrated activation global scales and dynamic block scales. Prefix caching is disabled. Peak memory is the maximum serving-phase NVML usage across repetitions.

| Format | Method | TTFT (ms) ↓ | Speedup ↑ | TPOT (ms) ↓ | Speedup ↑ | Output tok/s ↑ | Speedup ↑ | Peak mem. (GiB) ↓ | Reduction ↑ |
|---|---|---|---|---|---|---|---|---|---|
| **Batch 8** | | | | | | | | | |
| **MXFP4** | | | | | | | | | |
| BF16 | baseline | 926.50 | 1.00× | 14.14 | 1.00× | 502.1 | 1.00× | 21.87 | 1.00× |
| MXFP4 | RTN | 271.00 | 3.42× | 9.96 | 1.42× | 763.5 | 1.52× | 12.61 | 1.73× |
| | LoDiQ-Lite | 308.41 | 3.00× | 10.08 | 1.40× | 750.2 | 1.49× | 12.61 | 1.73× |
| | LoDiQ-Pro | 315.75 | 2.93× | 10.39 | 1.36× | 727.7 | 1.45× | 12.77 | 1.71× |
| | LoDiQ-Max | 315.75 | 2.93× | 10.40 | 1.36× | 728.2 | 1.45× | 12.89 | 1.70× |
| **NVFP4** | | | | | | | | | |
| BF16 | baseline | 1003.87 | 1.00× | 13.94 | 1.00× | 503.7 | 1.00× | 21.88 | 1.00× |
| NVFP4 | RTN | 323.98 | 3.10× | 10.77 | 1.29× | 702.4 | 1.39× | 12.82 | 1.71× |
| | LoDiQ-Lite | 319.40 | 3.14× | 10.97 | 1.27× | 691.1 | 1.37× | 12.83 | 1.71× |
| | LoDiQ-Pro | 346.42 | 2.90× | 11.83 | 1.18× | 640.8 | 1.27× | 13.10 | 1.67× |
| | LoDiQ-Max | 333.98 | 3.01× | 12.26 | 1.14× | 620.4 | 1.23× | 13.94 | 1.57× |
| **Batch 16** | | | | | | | | | |
| **MXFP4** | | | | | | | | | |
| BF16 | baseline | 2072.16 | 1.00× | 15.87 | 1.00× | 804.1 | 1.00× | 27.28 | 1.00× |
| MXFP4 | RTN | 627.14 | 3.30× | 11.69 | 1.36× | 1241.0 | 1.54× | 18.04 | 1.51× |
| | LoDiQ-Lite | 647.29 | 3.20× | 11.94 | 1.33× | 1213.8 | 1.51× | 18.04 | 1.51× |
| | LoDiQ-Pro | 663.96 | 3.12× | 12.21 | 1.30× | 1186.4 | 1.48× | 19.73 | 1.38× |
| | LoDiQ-Max | 666.06 | 3.11× | 12.23 | 1.30× | 1184.1 | 1.47× | 18.32 | 1.49× |
| **NVFP4** | | | | | | | | | |
| BF16 | baseline | 2045.01 | 1.00× | 15.85 | 1.00× | 807.3 | 1.00× | 27.28 | 1.00× |
| NVFP4 | RTN | 653.25 | 3.13× | 12.74 | 1.24× | 1143.4 | 1.42× | 18.25 | 1.49× |
| | LoDiQ-Lite | 696.09 | 2.94× | 12.81 | 1.24× | 1130.9 | 1.40× | 18.25 | 1.49× |
| | LoDiQ-Pro | 716.07 | 2.86× | 13.66 | 1.16× | 1063.9 | 1.32× | 20.01 | 1.36× |
| | LoDiQ-Max | 745.40 | 2.74× | 13.98 | 1.13× | 1038.1 | 1.29× | 19.37 | 1.41× |

Table 14: Distillation-data ablation on Qwen3-8B with LoDiQ-Max. Teacher-generated sequences start from a single-token prompt. Accuracy is reported in percent. BF16 is copied from the main results as a reference. Shading marks the default data source, and bold marks the best quantized result within each format.

| Format | Distillation data | Decode KLD ↓ | WT2 PPL ↓ | 0-shot Avg. ↑ | MMLU 5-shot ↑ |
|---|---|---|---|---|---|
| BF16 | – | – | 9.72 | 70.43 | 74.93 |
| MXFP4 | AceReason | **0.13968** | **9.7308** | 69.75 | 72.00 |
| | RedPajama | 0.15173 | 9.8203 | **69.99** | **72.70** |
| | Teacher-generated | 0.15294 | 9.9388 | 69.55 | 72.37 |
| | Random tokens | 0.18554 | 10.0458 | 68.38 | 71.36 |
| NVFP4 | AceReason | **0.08061** | **9.5584** | 70.35 | 73.47 |
| | RedPajama | 0.08683 | 9.7350 | 69.97 | 73.43 |
| | Teacher-generated | 0.08504 | 9.6396 | **70.37** | **73.79** |
| | Random tokens | 0.10440 | 9.9585 | 69.41 | 73.34 |

Table 15: Training-token ablation on Qwen3-8B with LoDiQ-Max. Accuracy is reported in percent. BF16 is the main-table reference. Shading marks the default 1M-token run. The best quantized result within each format is bolded.

| Format | Training tokens | Decode KLD ↓ | WT2 PPL ↓ | 0-shot Avg. ↑ | MMLU 5-shot ↑ |
|---|---|---|---|---|---|
| BF16 | – | – | 9.72 | 70.43 | 74.93 |
| MXFP4 | 1M | 0.139684 | 9.731 | 69.75 | 72.00 |
| | 5M | **0.136130** | 9.614 | 69.49 | 71.71 |
| | 10M | 0.142904 | **9.572** | **70.34** | **72.38** |
| NVFP4 | 1M | 0.080615 | 9.558 | **70.35** | **73.47** |
| | 5M | **0.077145** | 9.476 | 70.00 | 73.27 |
| | 10M | 0.078206 | **9.432** | 70.10 | 73.45 |

### A.4 Distillation Data and Training-Token Ablations

**Distillation data.** Table 14 compares distillation data sources for LoDiQ-Max on Qwen3-8B. AceReason gives the lowest WikiText-2 perplexity and decode KLD in both formats, but the highest task accuracy depends on the data source. RedPajama gives the highest zero-shot average and MMLU under MXFP4, while teacher-generated data gives the highest values under NVFP4. Random tokens yield higher perplexity and decode KLD than the other sources in both formats. The data-source comparison therefore favors meaningful input sequences without identifying a single source as best across all metrics.

**Training-token budget.** Table 15 reports the aggregate results for 1M, 5M, and 10M training tokens. WikiText-2 perplexity decreases with more training tokens in both formats. Decode KLD is lowest at 5M, and downstream accuracy is not uniformly monotonic. At 10M, MXFP4 reaches a zero-shot average of 70.34 and MMLU of 72.38, compared with 69.75 and 72.00 at 1M. NVFP4 already attains its highest zero-shot average and MMLU at 1M. These results support 1M tokens as a low-cost default without implying that additional training cannot improve individual metrics.