# Scaling Laws for Mixed Quantization

**Zeyu Cao[1]*, Yao Lai[1]*, Boyang Gu[2]*, Cheng Zhang[2], Pedro Gimenes[2]**
**Jianqiao Lu[3], Xitong Gao[4], Jianyi Cheng[5], Robert D. Mullins[1], Yiren Zhao[2]**
[1]University of Cambridge, [2]Imperial College London, [3]The University of Hong Kong,
[4]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[5]The University of Edinburgh
{zc317,yl2204}@cam.ac.uk, {boyang.gu19,a.zhao}@imperial.ac.uk

## Abstract

Post-training quantization (PTQ) of large language models (LLMs) reduces memory usage and accelerates inference while maintaining accuracy. However, it remains unclear how to select a quantization scheme that achieves target accuracy across different model scales. We introduce two parameters to characterize this: the quantization ratio $Q_r$ (proportion of parameters quantized to low precision) and the block size $Q_b$ (number of parameters sharing a scaling factor). Based on extensive experiments across multiple LLM families and scales, as well as diverse quantization schemes, we propose a unified scaling law that predicts quantization-induced loss degradation as a function of $Q_r$, $Q_b$, and model size. Our scaling law reveals two practical implications within the calibrated regime: (1) larger models tolerate higher $Q_r$, making mixed-precision increasingly effective at scale; (2) the absolute loss benefit of finer block granularity diminishes with model size, informing the trade-off between quantization error and hardware overhead. These findings provide a principled recipe for quantizing LLMs in practical deployment.

## 1 Introduction

The rapid scaling of large language models (LLMs) to hundreds of billions of parameters has made efficient deployment increasingly challenging (Anthropic, 2025; OpenAI, 2025; DeepMind, 2025). This growth has motivated research on scaling laws, which characterize the optimal trade-off between parameter count and training tokens under a fixed compute budget (Kaplan et al., 2020; Hoffmann et al., 2022).

However, existing scaling laws primarily characterize training-time compute optimality, leaving the inference and deployment side under-explored even as quantization becomes the dominant tool for reducing inference cost (Lin et al., 2024; Chee et al., 2024; Ashkboos et al., 2024; Zhu et al., 2024). Recent work on precision scaling (Kumar et al., 2024) models uniform-precision degradation, but how *mixed* post-training quantization (PTQ) interacts with model scale and configuration remains poorly understood.

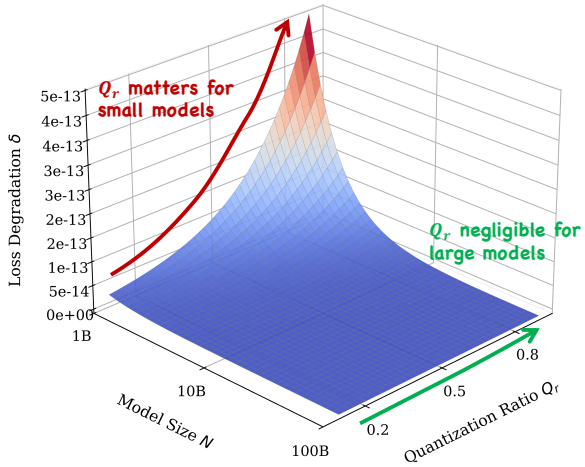



Figure 1: **Visualization of mixed-precision scaling law.** Loss degradation $\delta$ decreases with model size $N$, but increases with quantization ratio $Q_r$ . The fitted parameters are obtained from experiments on the Qwen-3 models with block size $Q_b = 32$.

In practice, LLM quantization often employs mixed-precision schemes rather than uniform low-bit arithmetic (Frantar et al., 2022), partly because pre-trained transformers exhibit magnitude outliers in weights and activations (Gu et al., 2024) that are commonly handled by assigning higher precision to outliers while quantizing the rest to lower precision (Dettmers et al., 2022; Zhang et al., 2023; Dettmers et al., 2023). Recent hardware advances further enable such strategies: NVIDIA's Blackwell GPUs introduce compact MXFP formats (4-bit and 6-bit) with block-wise scaling (Tirumala and Wong, 2024), where a block of $Q_b$ values shares a common scaling factor (Fig. 2), making both high- and low-precision arithmetic efficient on silicon. This motivates a scaling law for mixed-precision

*Equal contribution.

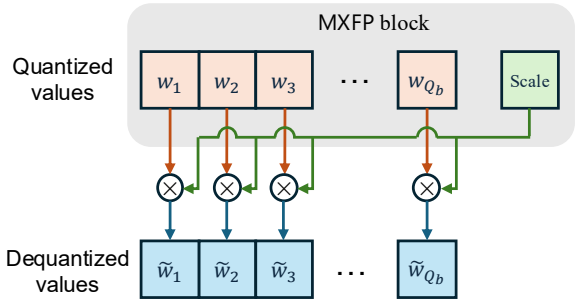



Figure 2: **MXFP block with $Q_b$ values sharing one scale.** A larger block size $Q_b$ shares one scale across more values, reducing scale overhead but typically increasing quantization error due to a coarser dynamic-range fit.

quantization.

This raises a largely unexplored question: *How should the ratio of low-precision elements and the block granularity in mixed quantization scale with model size?*

To make this question precise, we parametrize mixed quantization by two factors: the ratio of low-precision parameters ($Q_r$) and the block granularity ($Q_b$). We focus on the post-training quantization setting and evaluate across multiple LLM families, covering both weight-only and weight-activation quantization.

For a model with $N$ parameters trained on $D$ tokens, different assignments of low- and high-precision parameters induce different inference losses. We model this loss as the standard Chinchilla scaling form (Hoffmann et al., 2022) plus a quantization-induced degradation term $\delta$:

$$L(N, D, Q_r, Q_b) = \underbrace{aN^{-\alpha} + bD^{-\beta} + E}_{\text{Chinchilla form}} + \underbrace{\delta(N, Q_r, Q_b)}_{\text{Quantization loss}}, \tag{1}$$

where $\delta$ is treated as a random variable over the space of mixed-precision configurations satisfying $(Q_r, Q_b)$.

In this work, we find that both the optimal and expected quantization loss degradation follow a unified multiplicative scaling law:

$$\delta(N, Q_r, Q_b) = C \cdot \underbrace{e^{AQ_r}}_{\text{ratio}} \cdot \underbrace{N^{-\gamma_N}}_{\text{size}} \cdot \underbrace{(Q_b + d)^{\gamma_c}}_{\text{granularity}}, \tag{2}$$

where $A$, $\gamma_N$, $\gamma_c$, $d$, and $C$ are fitted constants. According to this scaling law, larger models exhibit smaller quantization degradation due to the $N^{-\gamma_N}$ term, enabling them to tolerate higher quantization ratios $Q_r$ without significant performance loss. Meanwhile, the absolute loss gap between block sizes shrinks with $N$, so finer granularity provides less absolute benefit for larger models under a fixed loss budget. The relative block-size effect in our separable form remains independent of $N$.

We make the following contributions:

- We propose a unified scaling law that models quantization loss as a function of model size, mixed-precision ratio, and block granularity.
- We prove our formulation recovers the Precision Scaling Law of Kumar et al. (2024) as a special case.
- We validate across 17 models (60M–8B for the main fits, with extrapolation to 32B in Appendix H) and multiple quantization methods over 115,500 checkpoints, and derive practical mixed-precision guidelines.

Our analysis reveals that the mixed-quantization ratio is the primary driver of quantization loss, while finer block granularity exhibits diminishing returns as model size increases.

## 2 Preliminary

### 2.1 Setup and Notation

Mixed-precision quantization assigns different precisions to different subsets of model parameters. We consider partitioning the weight matrices into a low-precision set $W_l$ (e.g., 4-bit) and a high-precision set $W_h$ (e.g., 16-bit), and define the mixed-quantization ratio as

$$Q_r(W_l, W_h) := \frac{\|W_l\|_0}{\|W_l\|_0 + \|W_h\|_0}, \tag{3}$$

where $\|\cdot\|_0$ denotes the number of parameters. The block granularity $Q_b$ is the number of values per quantization block that share a common scaling factor; smaller $Q_b$ enables finer-grained quantization but increases overhead. Given a target pair $(Q_r, Q_b)$, multiple valid partitions $(W_l, W_h)$ exist. Since different partitions induce different inference losses, we treat the quantization loss as a random variable over this configuration space.

### 2.2 Chinchilla and Precision Scaling Laws

The Chinchilla scaling law (Hoffmann et al., 2022) characterizes how language model loss scales with model size and training data. For a model with $N$ parameters trained on $D$ tokens, the loss $L$ follows

$$L(N, D) = aN^{-\alpha} + bD^{-\beta} + E\,, \tag{4}$$

where $a$, $b$, $\alpha$, $\beta$, and $E$ are positive constants.

The Precision Scaling Law (Kumar et al., 2024) extends this to post-training quantization. Let

$\mathbf{P}_{\text{train}} = (P_{\text{w}}, P_{\text{a}}, P_{\text{kv}})$ denote the training precisions for weights, activations, and KV-cache, respectively. When the model is trained at precision $\mathbf{P}_{\text{train}}$ but evaluated at a lower precision $P_{\text{post}}$, Eq. (4) becomes

$$L(N, D, \mathbf{P}_{\text{train}}, P_{\text{post}}) = aN_{\text{eff}}^{-\alpha} + bD^{-\beta} + E + \delta(N_{\text{eff}}, \mathbf{P}_{\text{train}}, P_{\text{post}}), \tag{5}$$

where $N_{\text{eff}}$ is the effective parameter count adjusted for training precision, and $\delta(\cdot)$ is the loss degradation induced by post-training quantization. Both terms are detailed in Kumar et al. (2024).

## 3 Mixed Quantization Scaling Laws

The Precision Scaling Law (Eq. (5)) assumes uniform precision across all weights. To extend to mixed-precision quantization, we introduce the quantization ratio $Q_r$ and block granularity $Q_b$ defined in Sec. 2; $Q_r$ acts as a *global budget* that allows per-layer heterogeneity through the partition search, with empirical support in Appendix B.1.

For fixed $(N, Q_r, Q_b)$, multiple valid partitions $(W_l, W_h)$ exist, each yielding a different quantization loss. We treat the partition as a random variable over the configuration space, so $\delta(N, Q_r, Q_b)$ becomes a random variable whose distribution depends on $(N, Q_r, Q_b)$. Accordingly, Eq. (5) becomes

$$L(N, D, Q_r, Q_b) = aN^{-\alpha} + bD^{-\beta} + E + \delta(N, Q_r, Q_b). \tag{6}$$

Since the first three terms are independent of the partition, minimizing $L$ reduces to minimizing $\delta$. We study two summary statistics of this distribution: (1) the optimal quantization loss $\delta^{\text{opt}}(N, Q_r, Q_b) = \min_{(W_l, W_h)} \delta$, representing the best achievable performance, and (2) the expected loss $\mathbb{E}[\delta(N, Q_r, Q_b)]$ over random partitions, representing average performance.

The number of candidate partitions grows combinatorially with model size, making exact optimization intractable. Instead, we uniformly sample partitions by randomly assigning a precision to each network component (*i.e.*, layer or weight matrix). We estimate $\mathbb{E}[\delta]$ via the sample mean $\delta_\mu$, and approximate $\delta^{\text{opt}}$ by the minimum over sampled partitions. Empirically, we find that even a modest number of samples suffices to characterize the distribution of $\delta$ (see Appendix B).

No fine-tuning is performed after quantization, so $\delta$ reflects the immediate degradation from post-training quantization. In Sec. 4, we show that both $\delta^{\text{opt}}(N, Q_r, Q_b)$ and $\mathbb{E}[\delta(N, Q_r, Q_b)]$ follow predictable scaling laws as functions of $N$, $Q_r$, and $Q_b$.

### 3.1 Proposed Scaling Law

We hypothesize that model size, quantization ratio, and block granularity affect the loss degradation independently:

- **Parameter scaling**: For fixed $Q_r$ and $Q_b$, $\delta$ decreases polynomially in number of parameters $N$.
- **Ratio scaling**: For fixed $N$ and $Q_b$, $\delta$ increases exponentially in $Q_r$.
- **Granularity scaling**: For fixed $N$ and $Q_r$, $\delta$ increases polynomially in $Q_b$.

Like prior language-model scaling laws, our functional form is an empirical hypothesis rather than a mechanistic derivation. Its structure is nevertheless motivated by two limiting cases and the observed log-linear trends: quantization errors accumulated across additional low-precision components approximately compound, making $\log \delta$ roughly linear in $Q_r$ and motivating $e^{AQ_r}$; the polynomial dependence on $N$ follows precision scaling. The multiplicative composition reduces to the precision scaling form at $Q_r = 0$ and to the weak law when $Q_b$ is fixed. The resulting separability is an approximation supported within our measured regime; explicit interaction terms remain an open direction.

Based on these observations, we propose two forms of the scaling law. We refer to these as the *weak* and *strong* forms: the weak form holds $Q_b$ fixed, while the strong form models the dependence on $Q_b$ explicitly. We validate this scaling law empirically in Sec. 4.

### 3.2 Weak Scaling Law

We first consider the case where block size $Q_b$ is fixed, and model the loss degradation as a function of model size $N$ and quantization ratio $Q_r$.

**Weak Scaling Law**

For a fixed quantization block size $Q_b$ and quantization method, the optimal and expected loss degradation scale as:

$$\delta^{\text{opt}}(N, Q_r) = Ce^{AQ_r}N^{-\gamma_N}\,, \tag{7}$$

$$\mathbb{E}[\delta(N, Q_r)] = C'e^{A'Q_r}N^{-\gamma'_N}\,, \tag{8}$$

where $C, C', A, A', \gamma_N, \gamma'_N$ are constants depending on the block size, quantization method, and model family.

We use *unified* throughout the paper to refer to the shared parametric form above: the same functional structure applies across quantization methods, model families, and granularities, while the coefficients themselves are setting-specific and should be calibrated per (model family, quantization method) pair. The practical value is therefore that of a sample-efficient calibration tool — fitting a small number of coefficients on a few small checkpoints, rather than a single universal predictor with family-agnostic constants.

We interpret $e^{AQ_r}$ and $N^{-\gamma_N}$ as the quantization scaling effect and parameter scaling effect, respectively. Our scaling law extends the precision scaling law in Eq. (5). When $Q_r = 0$, all weights use high precision, so the quantization scaling effect reduces to a constant absorbed into $C$. The parameter scaling effect $N^{-\gamma_N}$ is consistent with prior work (Kumar et al., 2024), which also observed polynomial dependence on model size.

The weak scaling law reveals a trade-off between model size and quantization ratio: while $\delta^{\text{opt}}$ increases exponentially with $Q_r$, a larger $N$ can compensate for this degradation. Fixing a loss budget $\delta^{\text{opt}} = l$, we obtain:

$$\log(N) = A''Q_r + C''\,, \tag{9}$$

where $A'' = A/\gamma_N$ and $C'' = (\log(C) - \log(l))/\gamma_N$. This shows that larger models can tolerate higher quantization ratios for the same loss budget, consistent with empirical findings in AWQ (Lin et al., 2024), QuIP (Chee et al., 2024), and LQER (Zhang et al., 2024). Equivalently, for a fixed quantization ratio, larger models suffer less degradation.

### 3.3 Strong Scaling Law

The weak scaling law suffices when only $N$ and $Q_r$ vary. To capture the effect of block granularity $Q_b$, we introduce the strong scaling law, where the granularity scaling effect is independent of $N$ and $Q_r$.

**Strong Scaling Law**

For a fixed quantization method, the optimal and expected loss degradation scale as:

$$\delta^{\text{opt}}(N, Q_r, Q_b) = Ce^{AQ_r}N^{-\gamma_N}(Q_b+d)^{\gamma_b}\,, \tag{10}$$

$$\mathbb{E}[\delta(N, Q_r, Q_b)] = C'e^{A'Q_r}N^{-\gamma'_N}(Q_b+d')^{\gamma'_b}\,, \tag{11}$$

where all coefficients depend on the quantization method.

When $Q_b$ is fixed, the granularity term $(Q_b + d)^{\gamma_b}$ becomes a constant and can be absorbed into $C$, reducing to the weak scaling law.

## 4 Experiments

### 4.1 Setup

**Models, data, and evaluation metrics.** We evaluate models from the LLaMA (Touvron et al., 2023; Dubey et al., 2024), Qwen-1.5 (Bai et al., 2023), and Qwen-3 (Yang et al., 2025) families, with sizes ranging from 0.5B to 8B for the main scaling-law fits; we additionally validate the fitted law on 13B–32B checkpoints in Appendix H. To enable extrapolation from small-scale models, we also pretrain a series of LLaMA-like models with $N \in \{60\text{M}, 200\text{M}, 400\text{M}, 600\text{M}, 1.1\text{B}\}$ parameters on FineWeb (Lozhkov et al., 2024), following the Chinchilla scaling law. We refer to these as the CLM series; details are in Appendix A. For evaluation, we subsample 1000 entries from SlimPajama (Soboleva et al., 2023), an open-source pretraining dataset. We report perplexity, the exponential of cross-entropy loss, on this data, as pretraining loss better reflects base model quality than downstream task performance (Dubey et al., 2024).

**Quantization methods.** To verify that our scaling laws generalize across data formats, we use MXINT (Rouhani et al., 2023b), a microscaling integer format, including MXINT-2 (weight-and-activation) and MXINT-4 (weight-and-activation and weight-only); HQQ (weight-only, 4-bit, block size 64) (Badri and Shaji, 2023), a calibration-free quantization method, as low-precision formats, with BF16 as the high-precision format; GPTQ (Frantar et al., 2022), a one-shot post-training weight quantization method based on approximate second-order information; Minifloat (Micikevicius et al., 2022), low-bit floating-point encodings that trade exponent and mantissa bits for efficiency, with BF16 as the high-precision format; MXFP (Rouhani et al., 2023a), a block

floating-point scheme that shares a scaling exponent across small groups of low-bit floats, with BF16 as the high-precision format.

**Mixed quantization strategy.** We perform quantization at two granularities: layer-wise and matmul-wise. In layer-wise quantization, the quantization ratio $Q_r$ is determined by the fraction of transformer layers cast to low precision. In matmul-wise quantization, we assign precision to each individual matrix multiplication. For each configuration, we draw 100 random allocations to estimate the distribution mean and its sample minimum; we do not claim that this finite random search finds the combinatorial global optimum. Throughout the experiments, $\delta^{\text{opt}}$ denotes this sample-minimum approximation. Appendix B shows that the layer-wise distribution estimates are stable from 100 to 1000 samples. We do not run the same convergence sweep for the larger matmul-wise space, although its consistently high fitting quality provides an empirical check on the resulting statistics. We evaluate $Q_r \in \{0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.975\}$. Note that each trial only involves post-training quantization (PTQ), with no additional training.

**Platform and GPU hours.** Experiments use a cluster of DGX A100 nodes (8×40GB), totaling ∼15k A100 GPU hours for PTQ search and an additional ∼1k H100 GPU hours for CLM pretraining.

### 4.2 Experimental Verification of Scaling Laws

We present layer-wise results for CLM, LLaMA, and Qwen-3, with fitted Qwen-3 parameters in Table 1; matrix-multiplication-wise results and Qwen-1.5 are deferred to Appendix D.

**Results for the weak scaling law.** Fig. 3 shows the actual and fitted $\delta^{\text{opt}}$ for layer-wise MXINT-4 quantization on CLM and Qwen-3. The contour shape matches between actual and fitted losses, with $R^2 = 0.98$. The scaling law also fits the expectation $\mathbb{E}[\delta]$, with $R^2 = 0.99$; corresponding figures and fitted parameters are in Appendix F. From Fig. 3, we make the following observation consistent with our weak law:

**Observation 1**

As $N$ increases, the quantization ratio $Q_r$ can be increased, which is in agreement with our combined ratio scaling and parameter scaling terms: $e^{AQ_r}N^{-\gamma_N}$.

**Results for the strong scaling law.** As stated in Sec. 5, recent LLM quantization methods adopt fine-grained quantization, meaning tensors are split into small blocks, quantized and then scaled individually. In this part, we empirically verify our unified scaling law by performing mixed-precision quantization search at block size $Q_b \in \{16, 32, 64, 128, 256\}$. We present experiments on the CLM series with matrix-multiplication-wise quantization for finer granularity. As shown in Fig. 4, the contour shape matches between actual and fitted $\delta^{\text{opt}}$, with $R^2 = 0.95$. The scaling law also fits the expectation $\mathbb{E}[\delta]$, with $R^2 = 0.98$; corresponding figures and fitted parameters are in Appendix F. From Fig. 4, we make the following observation consistent with our strong law:

**Observation 2**

When the quantization ratio $Q_r$ is low, employing a smaller block size $Q_b$ is more effective in minimizing performance degradation, which agrees with our combined ratio scaling and granularity scaling terms: $e^{AQ_r}(Q_b + d)^{\gamma_c}$.

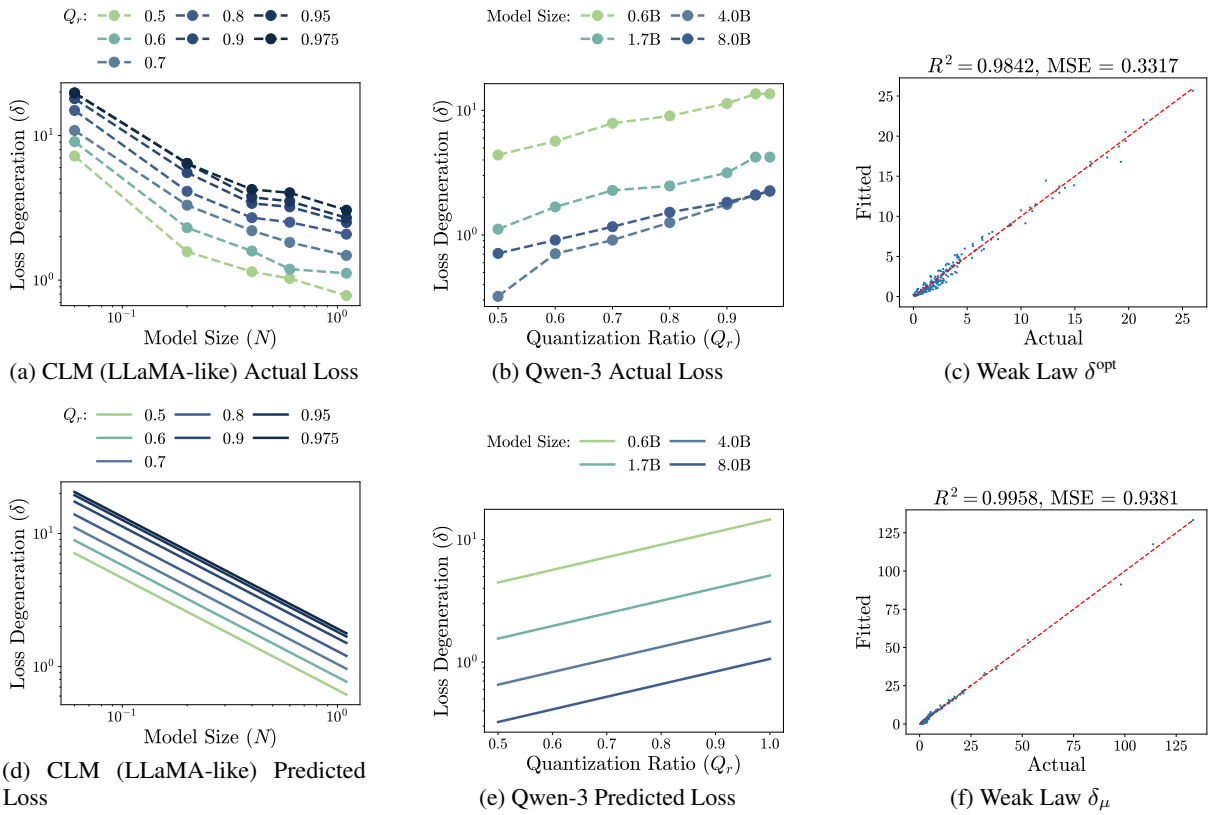
In the weak law $Q_b$ is fixed, so the granularity term $(Q_b + d)^{\gamma_c}$ is a constant folded into $C$. In the strong law it is factored out explicitly. Because $Q_r$, $Q_b$, and $N$ effects are partially entangled during fitting, separating the granularity term re-distributes weight onto the ratio coefficient $A$ and $\gamma_N$.

Table 1: Fitted scaling-law parameters for Qwen-3 under MXINT-4 weight-only quantization. We report both the weak and strong laws for $\delta^{\text{opt}}$ and $\delta_\mu$.

| Law | Target | $C$ | $A$ | $\gamma_N$ | $d$ | $\gamma_c$ |
|---|---|---|---|---|---|---|
| Weak | $\delta^{\text{opt}}$ | 0.0188 | 5.9150 | 1.1251 | – | – |
| Weak | $\delta_\mu$ | 0.3990 | 4.0875 | 1.9460 | – | – |
| Strong | $\delta^{\text{opt}}$ | 0.0025 | 8.1566 | 0.8181 | -13.1436 | 0.6208 |
| Strong | $\delta_\mu$ | 0.0142 | 4.1025 | 1.5909 | -3.7594 | 1.6736 |

### 4.3 More Quantization Methods

Our default setup uses W–A quantization with MXINT-4 (Sec. 4.1). We then test whether the proposed loss-degradation scaling law generalizes beyond this setting by varying (i) the numeric precision (MXINT-2 vs. MXINT-4), (ii) the quantization scheme (W–A vs. weight-only, with 16-bit activations), and (iii) the quantization method (MXINT vs. 4-bit HQQ). As shown in Fig. 5, the scaling law fits all three settings, but with different fitted parameters. Changing the precision (MXINT-4 to MXINT-2) leads to a substantial shift in the fitted parameters, unlike the behavior predicted by the




(a) CLM (LLaMA-like) Actual Loss (b) Qwen-3 Actual Loss (c) Weak Law $\delta^{\text{opt}}$

(d) CLM (LLaMA-like) Predicted Loss (e) Qwen-3 Predicted Loss (f) Weak Law $\delta_\mu$

Figure 3: **Weak Scaling Law Results.** (a, d): Loss degradation vs. model size $N$ for CLM, showing actual (a) and fitted (d) values. (b, e): Loss degradation vs. quantization ratio $Q_r$ for Qwen-3, showing actual (b) and fitted (e) values. (c, f): Fitted vs. actual values for $\delta^{\text{opt}}$ (c) and $\mathbb{E}[\delta]$ (f) across all MXINT-4 configurations (layer-wise and matmul-wise) on CLM, Qwen-1.5, and Qwen-3. The high $R^2$ values in (c, f) confirm that our weak scaling law accurately captures the empirical loss degradation. Full results are in Appendices C and D.

precision scaling law in Eq. (5). More results are deferred to Appendix E, including other formats' (GPTQ, Minifloat, MXFP) results.

**Observation 3**

Our unified scaling law can be applied to other quantization settings (e.g., WA vs. weight-only, and different numeric formats such as MXINT and HQQ).

### 4.4 Key Findings

As discussed in Sec. 2.2, our weak scaling law reduces to the precision scaling law when $Q_r = 0$. The strong scaling law, which adds a quasi-polynomial term in $Q_b$, remains consistent with the precision scaling law in the same limit.

**Takeaway 1**

Across different $Q_r$ values, our unified scaling law agrees asymptotically with the Precision Scaling Law of Kumar et al. (2024).

To examine the effect of $Q_r$ as model size $N$ varies, we plot Fig. 6a from our unified scaling law on the CLM series at a fixed $\delta$ budget. **Beyond 70B, $Q_r$ can exceed $0.9$ with loss degradation $\delta < 0.5$.**

**Takeaway 2**

With larger model size, the quantization ratio $Q_r$ can be pushed higher. This is because the parameter-scaling term $N^{-\gamma_N}$ in Eq. (2) shrinks faster with $N$ than the ratio-scaling term $e^{AQ_r}$ grows with $Q_r$.

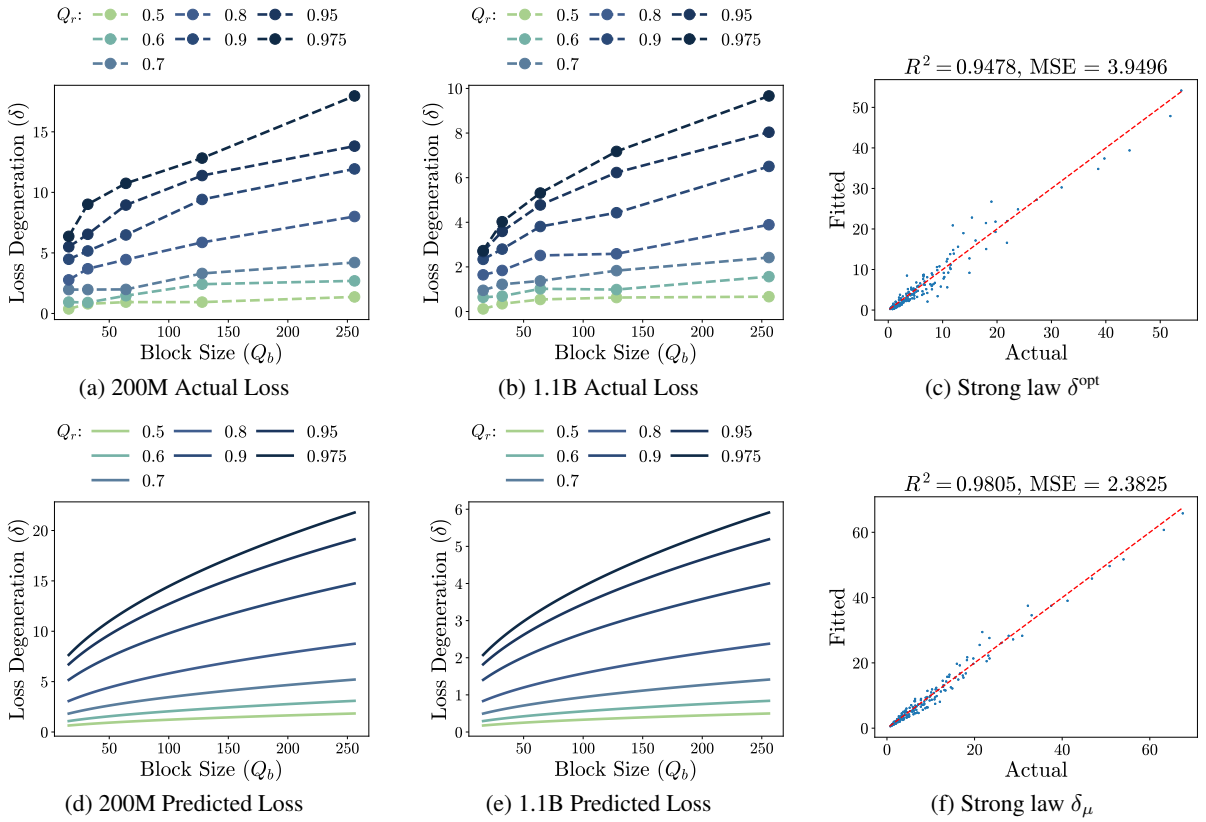
For different block-size granularities $Q_b$ as $N$ varies, we plot Fig. 6b from our unified scaling law on the CLM series. **For models larger than $50$B, the absolute gap between $Q_b = 128$ and $Q_b = 32$ in loss degradation $\delta$ is less than $0.5$.** Thus, under an absolute loss budget, finer blocks may offer limited benefit relative to their scaling-factor overhead in PTQ LLM inference.

**Takeaway 3**

As model size $N$ grows, finer block size $Q_b$ yields diminishing absolute loss benefits.

## 5 Related Work

**Quantization outliers and mixed quantization.** A weight or activation value is considered an out-



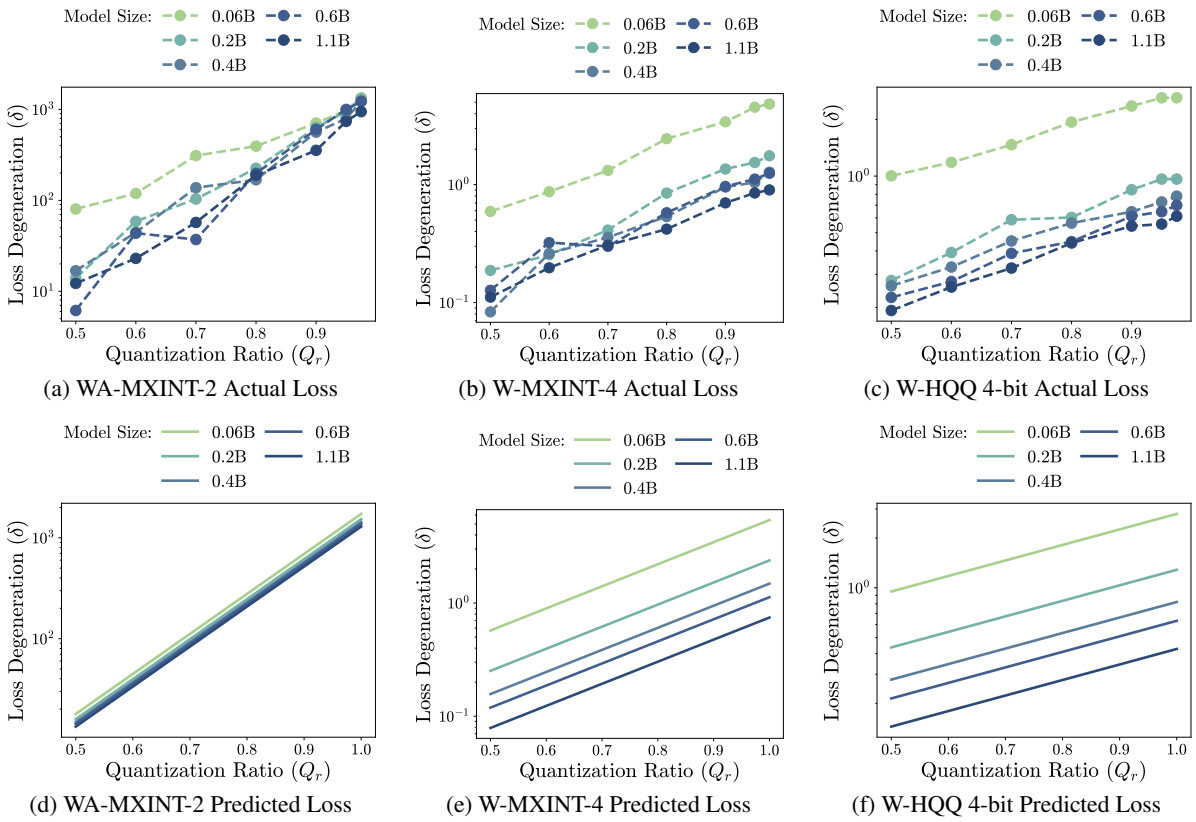

(a) 200M Actual Loss (b) 1.1B Actual Loss (c) Strong law $\delta^{\text{opt}}$

(d) 200M Predicted Loss (e) 1.1B Predicted Loss (f) Strong law $\delta_\mu$

Figure 4: **Strong Law Results**. (a,b,c,d): Actual and fitted loss degradation contours for CLM-200M and CLM-1.1B vs. block size $Q_b$. Each line corresponds to a fixed quantization ratio $Q_r$. (e,f): Fitted vs. actual $\delta^{\text{opt}}$ and $\delta_\mu$. We only show $N = 200$M, 1.1B due to space; see Appendix C for full results.

lier when there is a significant deviation from its mean distribution. Activation outliers have been observed more frequently in large models (Wei et al., 2022; Zhang et al., 2023) as cascaded layers accumulate quantization errors. In weight-only quantization, weights are mapped to low precision (Frantar et al., 2022; Lin et al., 2024). Recent weight-only quantization works focus on efficient vector quantization that maps high-precision weight tensors into indices and codebooks, such as QuiP (Chee et al., 2024), AQLM (Egiazarian et al., 2024), and QTIP (Tseng et al., 2024). Meanwhile, weight-activation quantization usually transfers activation magnitudes to weights using invertible scale matrices (Xiao et al., 2023) before quantizing both weights and activations (Wei et al., 2023; Xiao et al., 2023; Shao et al., 2023). Recent works explore incoherence processing to achieve this, such as SpinQuant (Liu et al., 2024b) and Quarot (Ashkboos et al., 2024). There are also works that solve the outlier problem via new number formats to accommodate the dynamic range of outliers (Zhang et al., 2023; Rouhani et al., 2023b; Zou et al., 2024). For example, MXINT/MXFP (Darvish Rouhani et al., 2020) is a recent standard for hardware-efficient numerical formats. MXINT shares an exponent across a block of mantissas (Rouhani et al., 2023b), and MXFP shares an exponent across a block of MiniFloats, which is already supported in NVIDIA Blackwell. The hardware efficiency of these methods often outperforms standard low-precision floating-point computation, although custom hardware support is required.

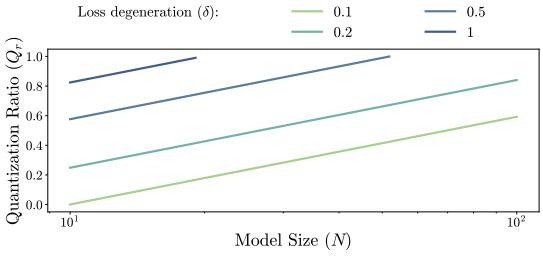
**Scaling laws of LLM training.** Training-time scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022; Pearce and Song, 2024) characterize how loss scales with parameter count and training tokens. A highly relevant follow-up is the *Scaling Laws for Precisions* (Kumar et al., 2024), which extends this to pre- and post-training quantization but assumes a single uniform precision level across all parameters. Closely related is recent work on scaling laws for quantization-aware training (QAT) (Frantar et al., 2025; Chen et al., 2025), which *incorporates* quantization effects during training or fine-tuning to improve low-precision performance. While complementary, we focus on PTQ. Beyond training-compute scaling, recent work also studies scaling under inference-time constraints: Sardana et al. (2023) extend Chinchilla-



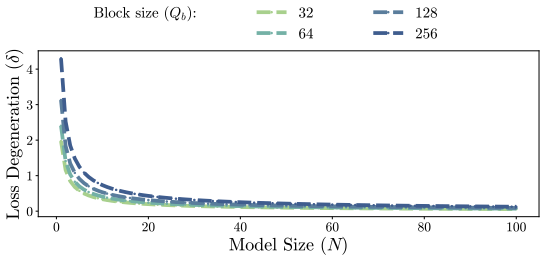

(a) WA-MXINT-2 Actual Loss (b) W-MXINT-4 Actual Loss (c) W-HQQ 4-bit Actual Loss

(d) WA-MXINT-2 Predicted Loss (e) W-MXINT-4 Predicted Loss (f) W-HQQ 4-bit Predicted Loss

Figure 5: **Other Arithmetic Formats Results.** (a,d) The actual and fitted loss degradation contour for layer-wise Weight-Activation-MXINT-2 (WA-MXINT-2) on CLM architectures. (b,e) The actual and fitted loss degradation contour for layer-wise Weight-only MXINT-4 (W-MXINT-4) on CLM architectures. (c,f) The actual and fitted loss degradation contour for layer-wise Weight-only 4bit-HQQ (HQQ-4) on CLM architectures. In the figure, each line corresponds to a fixed model size $N$ in billions (B). The fitted statistics are presented in Appendix F due to space constraints.




(a) $N$ compared to $Q_r$ under different fixed $\delta$ budget




(b) $N$ compared to $\delta$ under different fixed $Q_b$.

Figure 6: **Effects of our scaling laws.** The effect of our scaling law with (a) effect of quantization ratio $Q_r$ when changing model size $N$ under a fixed loss degradation budget $\delta$ and (b) effect of block size granularities $Q_b$ on loss degradation $\delta$ when changing $N$. Both are based on the fitted parameters for CLM series models.

style analysis to jointly account for inference cost when selecting model size and training tokens, and Bian et al. (2025) examine how architectural choices interact with scaling for inference-efficient LLMs. These directions are orthogonal to ours: rather than scaling pretraining compute or architecture, we model how the choice of mixed-precision configuration at deployment scales with model size. In Sec. 2.2, we further discuss the relevance of our mixed-quantization scaling laws, showing that in extreme cases our unified law reduces to the uniform-precision scaling law of Kumar et al. (2024).

## 6 Discussion

**Implications on AI inference hardware and systems.** We show that larger models can accommodate more low-precision components without performance degradation. This validates the recent trend of increasing support for low-precision arithmetic computation in hardware such as GPUs and TPUs (Tirumala and Wong, 2024; Choquette, 2022). The insight from our unified scaling law

highlights the need for increased low-precision resources in future hardware devices. In distributed inference (tensor/model parallel) and emerging heterogeneous serving, device-wise mixed precision becomes a new knob. Our law provides an accuracy-budget constraint to choose precision ratios without large-scale manual exploration.

**Further architectures and arithmetic formats.** Appendix H shows that our law fitted on $N \leq 8$B predicts PTQ loss with $R^2 \geq 0.96$ on 13–32B Qwen-1.5, Qwen-3, and LLaMA checkpoints; extending to 400B-scale dense LLMs, MoE models such as DeepSeek (Liu et al., 2024a) and Mixtral (Jiang et al., 2024), and further arithmetic formats (*e.g.*, ternary (Chen et al., 2024) and MXINT (Rouhani et al., 2023b)) remains open.

**Predictiveness and calibration.** Rather than a zero-shot predictor with universal coefficients, our law functions as a sample-efficient calibration tool, fitted on small checkpoints and extrapolated to larger ones in the same family.

## 7 Conclusion

We introduce a unified scaling law for post-training mixed quantization of LLMs, calibrated on small checkpoints and extrapolating across model size, ratio, and granularity. Our findings guide practical mixed-precision design and motivate broader low-precision support in inference hardware.

## Limitations

Our law is a sample-efficient calibration tool rather than a zero-shot predictor: the functional form is shared, but its coefficients must be re-fit for each (model family, quantization method) pair, following the convention of prior scaling-law studies. Our main conclusions concern approximately 4-bit mixed-precision PTQ. MXINT-2 weight–activation quantization is an extreme-compression stress test and produces unstable coefficients; partially distilled checkpoints can also exhibit negative measured degradation when quantization improves evaluation loss, violating our non-negativity assumption. These cases should therefore be diagnosed through small-model calibration rather than covered by the positive-scaling regime by default. Quantization-aware training and broader downstream-task evaluation are left to future work. Our work does not discuss the potential impact of quantization on model safety (Gonçalves and Strubell, 2023; Jaiswal et al., 2024; Xu et al., 2024; Marcuzzi et al., 2026; Hong et al., 2024). The study focuses mainly on English language modeling, leaving open whether the findings generalize to multilingual or non-English settings, since prior work has shown that quantization can affect languages non-uniformly, with potentially larger degradation for languages using non-Latin scripts (Marchisio et al., 2024).

**License and Intended Use.** All models and datasets used in this work are publicly available under licenses permitting academic research, consistent with our usage.

## References


Anthropic. 2025. Claude 3.7 sonnet system card. Technical report, Anthropic.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. 2024. Quarot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 37:100213–100240.

Hicham Badri and Appu Shaji. 2023. Half-quadratic quantization of large machine learning models.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Song Bian, Tao Yu, Shivaram Venkataraman, and Youngsuk Park. 2025. Scaling laws meet model architecture: Toward inference-efficient llms. *arXiv preprint arXiv:2510.18245*.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M De Sa. 2024. Quip: 2-bit quantization of large language models with guarantees. *Advances in Neural Information Processing Systems*, 36.

Mengzhao Chen, Chaoyi Zhang, Jing Liu, Yutao Zeng, Zeyue Xue, Zhiheng Liu, Yunshui Li, Jin Ma, Jie Huang, Xun Zhou, et al. 2025. Scaling law for quantization-aware training. *arXiv preprint arXiv:2505.14302*.

Tianqi Chen, Zhe Li, Weixiang Xu, Zeyu Zhu, Dong Li, Lu Tian, Emad Barsoum, Peisong Wang, and Jian Cheng. 2024. Ternaryllm: Ternarized large language model. *arXiv preprint arXiv:2406.07177*.

Jack Choquette. 2022. Nvidia hopper gpu: Scaling performance. In *2022 IEEE Hot Chips 34 Symposium (HCS)*, pages 1–46. IEEE Computer Society.

Bita Darvish Rouhani, Daniel Lo, Ritchie Zhao, Ming Liu, Jeremy Fowers, Kalin Ovtcharov, Anna Vinogradsky, Sarah Massengill, Lita Yang, Ray Bittner,

et al. 2020. Pushing the limits of narrow precision inferencing at cloud scale with microsoft floating point. *Advances in neural information processing systems*, 33:10271–10281.

Google DeepMind. 2025. Gemini 2.5 pro preview. Product page.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. GPT-3.int8(): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35:30318–30332.

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. 2023. Spqr: A sparse-quantized representation for near-lossless llm weight compression. *arXiv preprint arXiv:2306.03078*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The LLaMA 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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

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

Elias Frantar, Utku Evci, Wonpyo Park, Neil Houlsby, and Dan Alistarh. 2025. Compression scaling laws: Unifying sparsity and quantization. *arXiv preprint arXiv:2502.16440*.

Gustavo Gonçalves and Emma Strubell. 2023. Understanding the effect of model compression on social bias in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2663–2675.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. 2024. When attention sink emerges in language models: An empirical view. *arXiv preprint arXiv:2410.10781*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Junyuan Hong, Jinhao Duan, Chenhui Zhang, Zhangheng Li, Chulin Xie, Kelsey Lieberman, James Diffenderfer, Brian Bartoldson, Ajay Jaiswal, Kaidi Xu, et al. 2024. Decoding compressed trust: Scrutinizing the trustworthiness of efficient llms under compression. *arXiv preprint arXiv:2403.15447*.

Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Xianglong Liu, Luca Benini, Michele Magno, and Xiaojuan Qi. 2024. Slim-llm: Salience-driven mixed-precision quantization for large language models. *arXiv preprint arXiv:2405.14917*.

Ajay Jaiswal, Zhe Gan, Xianzhi Du, Bowen Zhang, Zhangyang Wang, and Yinfei Yang. 2024. Compressing llms: The truth is rarely pure and never simple. In *International Conference on Learning Representations*, volume 2024, pages 36973–36993.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Tanishq Kumar, Zachary Ankner, Benjamin F Spector, Blake Bordelon, Niklas Muennighoff, Mansheej Paul, Cengiz Pehlevan, Christopher Ré, and Aditi Raghunathan. 2024. Scaling laws for precision. *arXiv preprint arXiv:2411.04330*.

Shiyao Li, Xuefei Ning, Ke Hong, Tengxuan Liu, Luning Wang, Xiuhong Li, Kai Zhong, Guohao Dai, Huazhong Yang, and Yu Wang. 2023. Llm-mq: Mixed-precision quantization for efficient llm deployment. In *The Efficient Natural Language and Speech Processing Workshop with NeurIPS*, volume 9.

Wanchao Liang, Tianyu Liu, Less Wright, Will Constable, Andrew Gu, Chien-Chin Huang, Iris Zhang, Wei Feng, Howard Huang, Junjie Wang, Sanket Purandare, Gokul Nadathur, and Stratos Idreos. 2025. Torchtitan: One-stop pytorch native solution for production ready LLM pretraining. In *The Thirteenth International Conference on Learning Representations*.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen

Blankevoort. 2024b. Spinquant: Llm quantization with learned rotations. *arXiv preprint arXiv:2405.16406*.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. Fineweb-edu: the finest collection of educational content.

Kelly Marchisio, Saurabh Dash, Hongyu Chen, Dennis Aumiller, Ahmet Üstün, Sara Hooker, and Sebastian Ruder. 2024. How does quantization affect multilingual llms? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15928–15947.

Federico Marcuzzi, Xuefei Ning, Roy Schwartz, and Iryna Gurevych. 2026. How quantization shapes bias in large language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 363–404.

Naveen Mellempudi, Sudarshan Srinivasan, Dipankar Das, and Bharat Kaul. 2019. Mixed precision training with 8-bit floating point. *arXiv preprint arXiv:1905.12334*.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. 2017. Mixed precision training. *arXiv preprint arXiv:1710.03740*.

Paulius Micikevicius, Dusan Stosic, Neil Burgess, Marius Cornea, Pradeep Dubey, Richard Grisenthwaite, Sangwon Ha, Alexander Heinecke, Patrick Judd, John Kamalu, et al. 2022. Fp8 formats for deep learning. *arXiv preprint arXiv:2209.05433*.

OpenAI. 2025. Openai gpt-4.5 system card. Technical report, OpenAI.

Yeonhong Park, Jake Hyun, SangLyul Cho, Bonggeun Sim, and Jae W Lee. 2024. Any-precision llm: Low-cost deployment of multiple, different-sized llms. *arXiv preprint arXiv:2402.10517*.

Tim Pearce and Jinyeop Song. 2024. Reconciling kaplan and chinchilla scaling laws. *Preprint*, arXiv:2406.12907.

BD Rouhani, N Garegrat, T Savell, R Zhao, and A More. 2023a. Ocp microscaling formats (mx) specification. *Open Compute Project*.

Bita Darvish Rouhani, Ritchie Zhao, Venmugil Elango, Rasoul Shafipour, Mathew Hall, Maral Mesmakhosroshahi, Ankit More, Levi Melnick, Maximilian Golub, Girish Varatkar, et al. 2023b. With shared microexponents, a little shifting goes a long way. In *Proceedings of the 50th Annual International Symposium on Computer Architecture, ISCA*, pages 83:1–83:13.

Nikhil Sardana, Jacob Portes, Sasha Doubov, and Jonathan Frankle. 2023. Beyond chinchilla-optimal: Accounting for inference in language model scaling laws. *arXiv preprint arXiv:2401.00448*.

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

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

Xiao Sun, Jungwook Choi, Chia-Yu Chen, Naigang Wang, Swagath Venkataramani, Vijayalakshmi Viji Srinivasan, Xiaodong Cui, Wei Zhang, and Kailash Gopalakrishnan. 2019. Hybrid 8-bit floating point (hfp8) training and inference for deep neural networks. *Advances in neural information processing systems*, 32.

Ajay Tirumala and Raymond Wong. 2024. Nvidia blackwell platform: Advancing generative ai and accelerated computing. *2024 IEEE Hot Chips 36 Symposium*, pages 1–33.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Albert Tseng, Qingyao Sun, David Hou, and Christopher M De Sa. 2024. Qtip: Quantization with trellises and incoherence processing. *Advances in Neural Information Processing Systems*, 37:59597–59620.

Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. 2023. Outlier suppression+: Accurate quantization of large language models by equivalent and optimal shifting and scaling. *arXiv preprint arXiv:2304.09145*.

Xiuying Wei, Yunchen Zhang, Xiangguo Zhang, Ruihao Gong, Shanghang Zhang, Qi Zhang, Fengwei Yu, and Xianglong Liu. 2022. Outlier suppression: Pushing the limit of low-bit transformer language models. *Advances in Neural Information Processing Systems*, 35:17402–17414.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pages 38087–38099. PMLR.

Zhichao Xu, Ashim Gupta, Tao Li, Oliver Bentham, and Vivek Srikumar. 2024. Beyond perplexity: Multi-dimensional safety evaluation of llm compression. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15359–15396.

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

Shulin Zeng, Jun Liu, Guohao Dai, Xinhao Yang, Tianyu Fu, Hongyi Wang, Wenheng Ma, Hanbo Sun, Shiyao Li, Zixiao Huang, et al. 2024. Flightllm: Efficient large language model inference with a complete mapping flow on fpgas. In *Proceedings of the 2024 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, pages 223–234.

Cheng Zhang, Jianyi Cheng, George A Constantinides, and Yiren Zhao. 2024. Lqer: Low-rank quantization error reconstruction for llms. *arXiv preprint arXiv:2402.02446*.

Cheng Zhang, Jianyi Cheng, Ilia Shumailov, George A Constantinides, and Yiren Zhao. 2023. Revisiting block-based quantisation: What is important for sub-8-bit llm inference? *arXiv preprint arXiv:2310.05079*.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2024. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics (TACL)*, 12:1556–1577.

Lancheng Zou, Wenqian Zhao, Shuo Yin, Chen Bai, Qi Sun, and Bei Yu. 2024. Bie: Bi-exponent block floating-point for large language models quantization. In *Forty-first International Conference on Machine Learning*.

## A Pre-training CLM series Model

As discussed in Sec. 4.1, we pre-trained a series of causal language models (CLMs) to facilitate our derivation of mixed quantization scaling laws and corresponding experiments, before we cross-validate our scaling laws on other open-source models.

**Model architecture and tokenizer** We adopt the Llama-3 architecture, which incorporates group query attention (GQA) and rotary positional embedding (RoPE) (Dubey et al., 2024). Table 2 summarizes the model architecture of our CLM series. We use `HuggingFaceTB/cosmo2-tokenizer` [1], an open-source tokenizer trained on 1M tokens and released by HuggingFace.

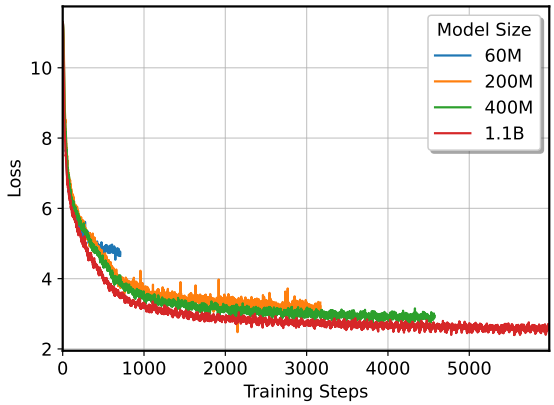



Figure 7: Pre-training loss of CLM series.

**Pre-training setup** We pre-train our models on FineWeb-Edu (Lozhkov et al., 2024), a pre-training dataset of cleaned and deduplicated english web data from CommonCrawl. We follow Chinchilla (Hoffmann et al., 2022) to determine the number of tokens for pre-training, *i.e.*, multiplying the number of parameters (without embedding layers) by 22. We use the AdamW optimizer with a linear learning rate schedule with 20% warmup. The initial learning rate is set to 1e-4 and the sequence length is set to 2048. We use various batch sizes for different models, *i.e.*, 96 for 60M, 64 for 200M, 96 for 400M, 96 for 600M, and 192 for 1.1B. Fig. 7 shows the pre-training loss of our CLM series. We plan to release our pre-trained models and tokenizer publicly.

**Pretraining framework** We use TorchTitan (Liang et al., 2025), a PyTorch-based distributed training framework to pretrain our models. We apply fully sharded data parallel (FSDP) and gradient checkpointing to save GPU memory. Eight NVIDIA H200 GPUs were used for pre-training 600M and 1.1B models, and eight NVIDIA A100 GPUs were used for pre-training 60M, 200M, and 400M models. Before conducting the mixed quantization experiments, we convert the pre-trained models from torch distributed format to HuggingFace format.

[1] `cosmo2-tokenizer`: https://huggingface.co/HuggingFaceTB/cosmo2-tokenizer

## B Distribution for Search Space

Fig. 8 presents the kernel density estimates (KDEs) of the distribution $\Delta$ under MXINT-4 layerwise quantization, evaluated across varying numbers of trials: 100, 200, 500, and 1000. These KDEs are generated using a Gaussian kernel, which may not be the optimal kernel, providing a consistent and smooth estimate that is sufficient for comparative analysis.

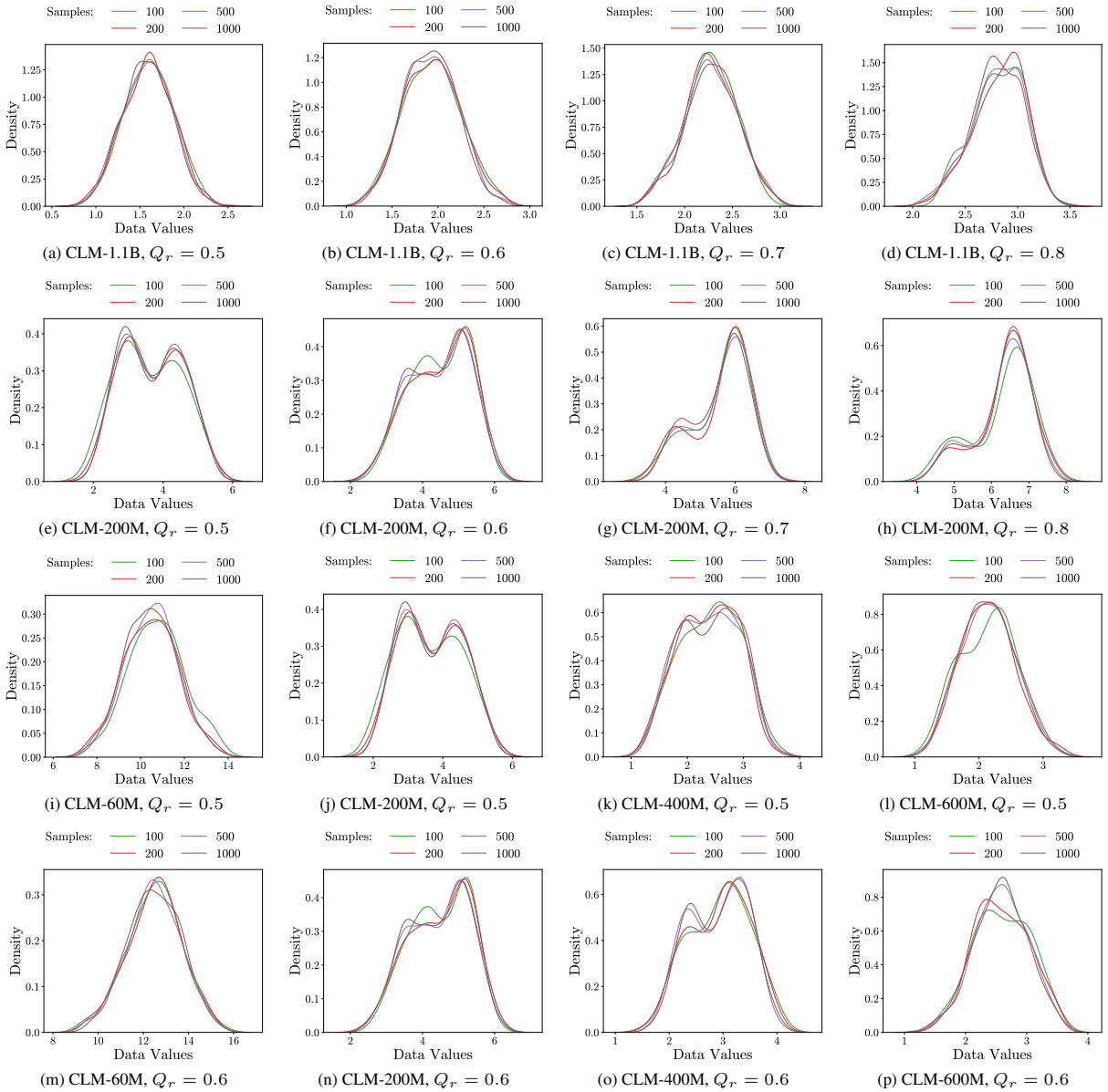
We present Figs. 8a to 8h, which show the KDEs of different quantization ratios $Q_r$ (from 0.5 to 0.8) for a fixed model (CLM-1.1B and CLM-200M), whereas Figs. 8i to 8p show the KDEs of different model sizes (from 60M to 600M) under fixed $Q_r$ (0.5 and 0.6). Across all figures, as the sample size increases from 100 to 1000, the estimated distribution tends to remain the same. The KDEs generated appear nearly indistinguishable in shape from that of 100. This consistency holds throughout different $Q_r$ and model sizes.

*These figures suggest that the empirical distribution of $\Delta$ can be reliably estimated with as few as 100 samples*, as presented in Sec. 4.1. That is, even with relatively small sample sizes, the mean ($\mathbb{E}(\Delta)$) and minimum ($\delta^{\text{opt}}$) values of $\Delta$ can still be well captured. Therefore, from both a computational and statistical efficiency perspective, excessive sampling provides diminishing returns. This insight is particularly valuable since large-scale sampling is computationally expensive.

### B.1 Layer-wise Sensitivity Heterogeneity Across Models

While our scaling laws use a single global quantization ratio $Q_r$ as a budget, the underlying search (described above) freely assigns different precisions to different layers. A natural question is whether the resulting layer-wise quantization preferences are similar across model sizes — if so, an explicit per-layer variable might improve the law.

To probe this, for each fixed $Q_r$ in the layer-wise




(a) CLM-1.1B, $Q_r = 0.5$ (b) CLM-1.1B, $Q_r = 0.6$ (c) CLM-1.1B, $Q_r = 0.7$ (d) CLM-1.1B, $Q_r = 0.8$

(e) CLM-200M, $Q_r = 0.5$ (f) CLM-200M, $Q_r = 0.6$ (g) CLM-200M, $Q_r = 0.7$ (h) CLM-200M, $Q_r = 0.8$

(i) CLM-60M, $Q_r = 0.5$ (j) CLM-200M, $Q_r = 0.5$ (k) CLM-400M, $Q_r = 0.5$ (l) CLM-600M, $Q_r = 0.5$

(m) CLM-60M, $Q_r = 0.6$ (n) CLM-200M, $Q_r = 0.6$ (o) CLM-400M, $Q_r = 0.6$ (p) CLM-600M, $Q_r = 0.6$

Figure 8: **KDE of distribution $\delta$.** Different KDE for different $Q_r$ under a fixed model: (a,b,c,d) Changing $Q_r$ for CLM-1.1B; (e,f,g,h) Changing $Q_r$ for CLM-200M. KDE for different model under fixed $Q_r$; (i,j,k,l) Changing $N$ for $Q_r = 0.5$; (m,n,o,p) Changing $N$ for $Q_r = 0.6$.

Table 2: Model architecture of CLMs.

| Model | Vocab size | Model dim | FFN dim | #Hidden layers | #Attention heads | #Key/value heads |
|---|---|---|---|---|---|---|
| 60M | | 384 | 1408 | 22 | 4 | 2 |
| 200M | | 768 | 2688 | 24 | 12 | 4 |
| 400M | 49152 | 960 | 3328 | 30 | 15 | 5 |
| 600M | | 1152 | 4096 | 32 | 18 | 6 |
| 1.1B | | 1536 | 5376 | 32 | 24 | 8 |

MXINT-4 setting we rank layers by their estimated quantization preference within a model, and compute the Spearman rank correlation of these per-layer rankings between every pair of CLM model sizes ($N \in \{60\text{M}, 200\text{M}, 400\text{M}, 600\text{M}, 1.1\text{B}\}$). We then average over the 10 model-size pairs at each $Q_r$ to obtain a single similarity score. The results are summarized in Table 3.

Table 3: Mean Spearman rank correlation of per-layer quantization preferences across pairs of CLM model sizes under layer-wise MXINT-4 quantization. Lower values indicate that the best per-layer precision assignment differs more substantially across model sizes.

| $Q_r$ | Mean Spearman |
|---|---|
| 0.5 | 0.5649 |
| 0.6 | 0.5953 |
| 0.7 | 0.5825 |
| 0.8 | 0.6041 |
| 0.9 | 0.4102 |

The mean correlation stays in the 0.41–0.60 range across $Q_r$, indicating that per-layer preferences are largely *model-specific* rather than transferable. This justifies our choice to keep $Q_r$ as a compact global budget rather than introducing per-layer variables: the latter would not generalize across models and would substantially complicate the mathematical form of the law without a corresponding gain in predictive utility for cross-model extrapolation.

## C Layer-wise Results

As shown in Sec. 4, Fig. 9 and Fig. 10 show the actual and fitted loss contour of CLM, Qwen-1.5, and Qwen-3 under MXINT-4 layerwise quantization. For each model family we present two contours: loss versus model size $N$ and loss versus quantization ratio $Q_r$. Fig. 9 shows the contour for the minimum value $\delta^{\text{opt}}$ and Fig. 10 shows the expectation value $\delta_\mu$. There are a few outliers that do not match our fitted contour, which could be caused by the instability of the sampling process. In addition, note that the actual losses themselves are estimations.

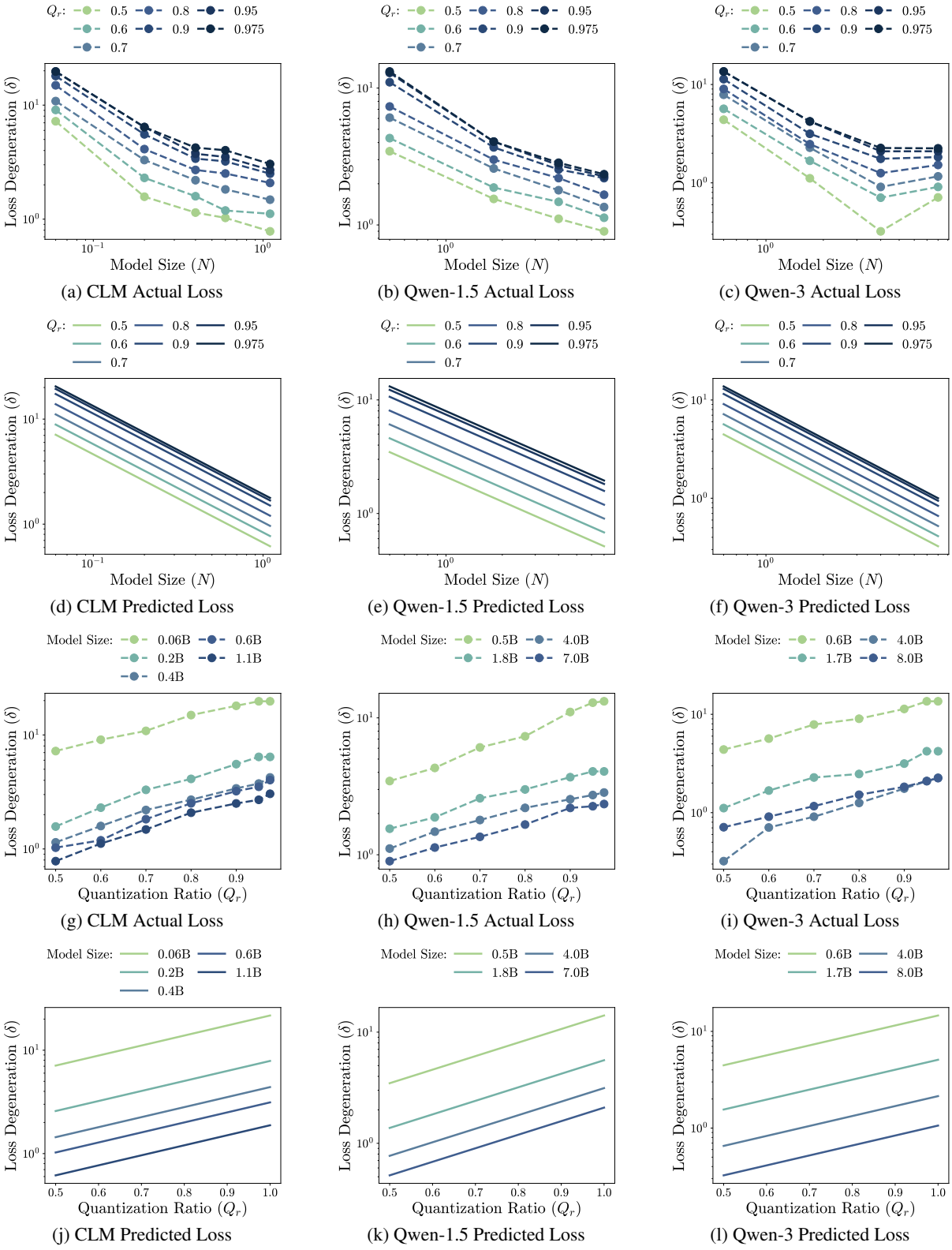
## D Matrix Multiplication-wise Results

Following the layer-wise results in Sec. 4, the matrix multiplication-wise with Fig. 11 and Fig. 12 shows the actual and fitted loss contour of CLM, Qwen-1.5, and Qwen-3 under MXINT-4 matrix multiplication-wise quantization. For each model family, we present two contours: loss versus model size $N$ and loss versus quantization ratio $Q_r$. Fig. 11 shows the contour for the minimum value $\delta^{\text{opt}}$ and Fig. 12 shows the expectation value $\delta_\mu$. As with the layer-wise quantization results, a few outliers deviate from the fitted contours due to the variability in the sampling process.

While Fig. 11 and Fig. 12 show the effectiveness of the weak law, Fig. 13, Fig. 14, and Fig. 15 demonstrate the utility for the strong law. Those three figures show the actual and fitted loss contours (both minimum and mean) with respect to block size $Q_b$ with different quantization ratios $Q_r$ for CLM, Qwen-1.5, and Qwen-3. For those figures, our strong law only takes into consideration losses that are less than 100, since larger losses in perplexity are meaningless under our setting (in practical terms, these models would not generate anything distinguishable). As a result, the fitted contours are not accurate when the actual loss is large, which is acceptable since we do not care about those excessive losses.

## E Other Numerical Format Results

Figs. 16 and 17 present the minimum and mean loss contours with respect to model size $N$ and quantization ratio $Q_r$ for the CLM, Qwen-1.5, and Qwen-3 models under HQQ quantization. These results demonstrate that HQQ is a highly effective quantization method, successfully preserving the capabilities of the pre-trained models and maintaining low post-training quantization (PTQ) loss. Notably, the fitted contours closely match the empirical data across all model families, indicating that our scaling law formulation generalizes well,



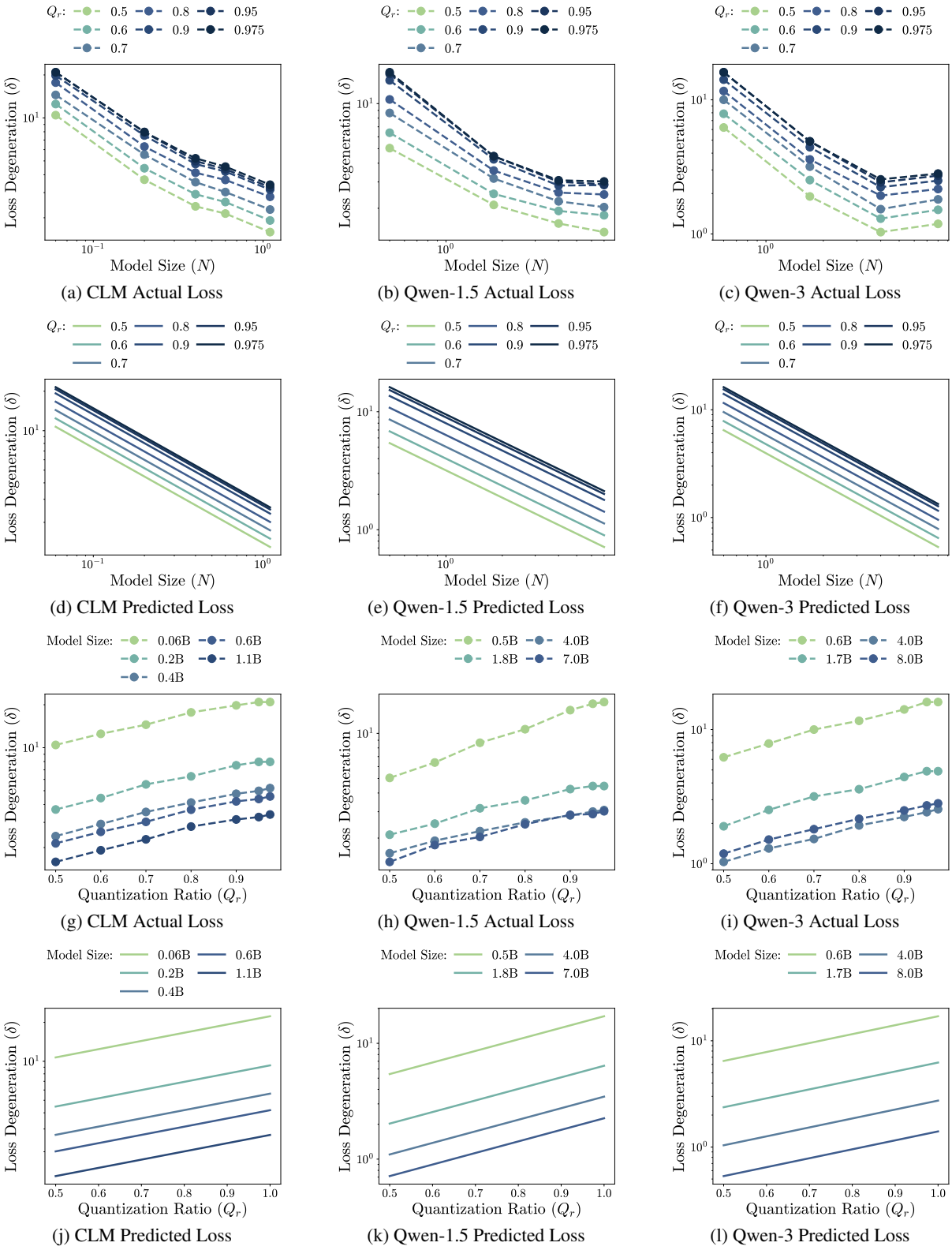

(a) CLM Actual Loss (b) Qwen-1.5 Actual Loss (c) Qwen-3 Actual Loss

(d) CLM Predicted Loss (e) Qwen-1.5 Predicted Loss (f) Qwen-3 Predicted Loss

(g) CLM Actual Loss (h) Qwen-1.5 Actual Loss (i) Qwen-3 Actual Loss

(j) CLM Predicted Loss (k) Qwen-1.5 Predicted Loss (l) Qwen-3 Predicted Loss

Figure 9: **Layer-wise** ($\delta^{\text{opt}}$) (a,d,g,h) CLM layer-wise results; (b,e,h,k) Qwen-1.5 layer-wise results; (c,f,i,l) Qwen-3 layer-wise results.



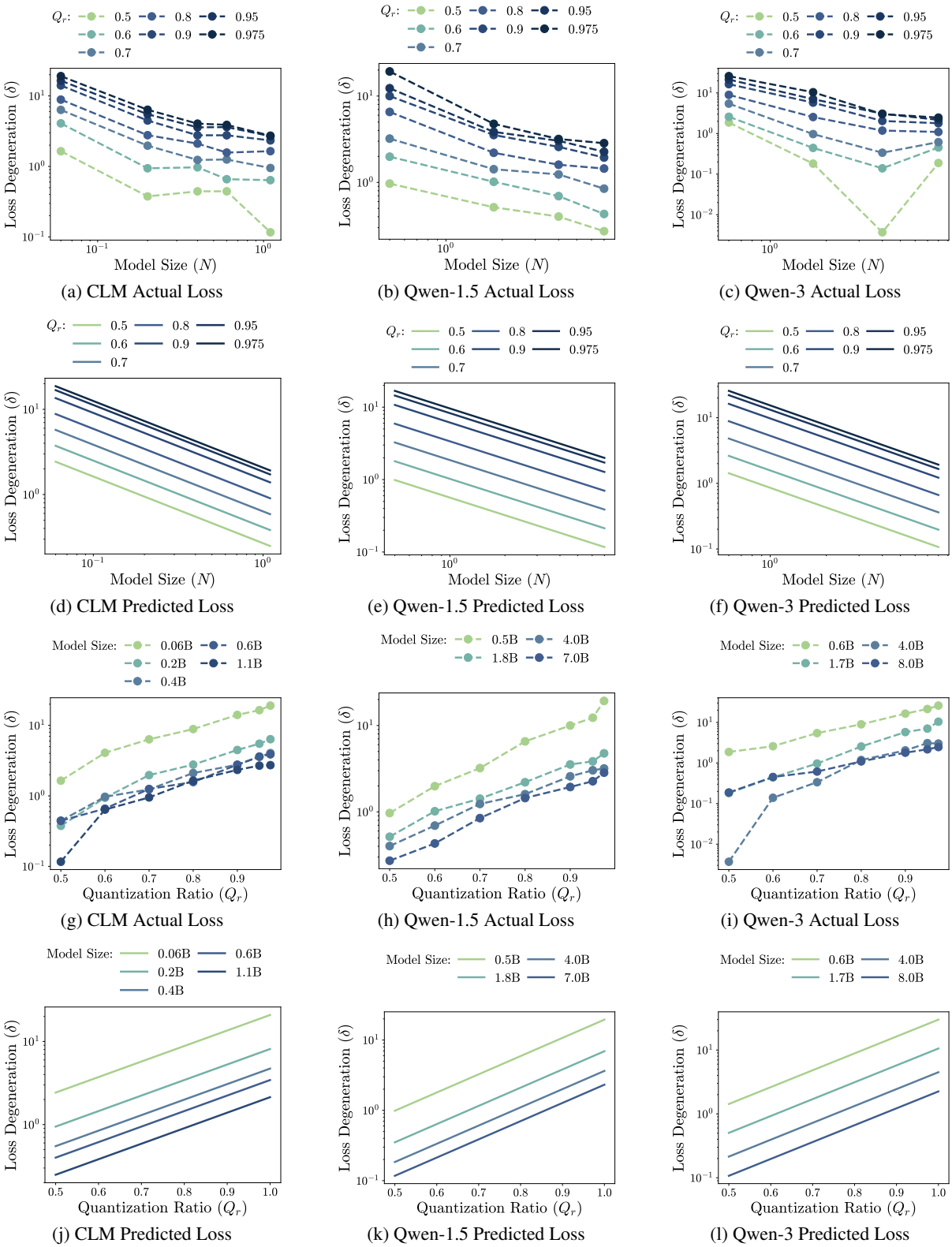

(a) CLM Actual Loss (b) Qwen-1.5 Actual Loss (c) Qwen-3 Actual Loss

(d) CLM Predicted Loss (e) Qwen-1.5 Predicted Loss (f) Qwen-3 Predicted Loss

(g) CLM Actual Loss (h) Qwen-1.5 Actual Loss (i) Qwen-3 Actual Loss

(j) CLM Predicted Loss (k) Qwen-1.5 Predicted Loss (l) Qwen-3 Predicted Loss

Figure 10: **Layer-wise ($\delta_\mu$)** (a,d,g,h) CLM layer-wise results; (b,e,h,k) Qwen-1.5 layer-wise results; (c,f,i,l) Qwen-3 layer-wise results.




(a) CLM Actual Loss (b) Qwen-1.5 Actual Loss (c) Qwen-3 Actual Loss

(d) CLM Predicted Loss (e) Qwen-1.5 Predicted Loss (f) Qwen-3 Predicted Loss

(g) CLM Actual Loss (h) Qwen-1.5 Actual Loss (i) Qwen-3 Actual Loss

(j) CLM Predicted Loss (k) Qwen-1.5 Predicted Loss (l) Qwen-3 Predicted Loss

Figure 11: **Matrix Multiplication-wise ($\delta^{\text{opt}}$)** (a,d,g,h) CLM matrix multiplication-wise results; (b,e,h,k) Qwen-1.5 matrix multiplication-wise results; (c,f,i,l) Qwen-3 matrix multiplication-wise results.

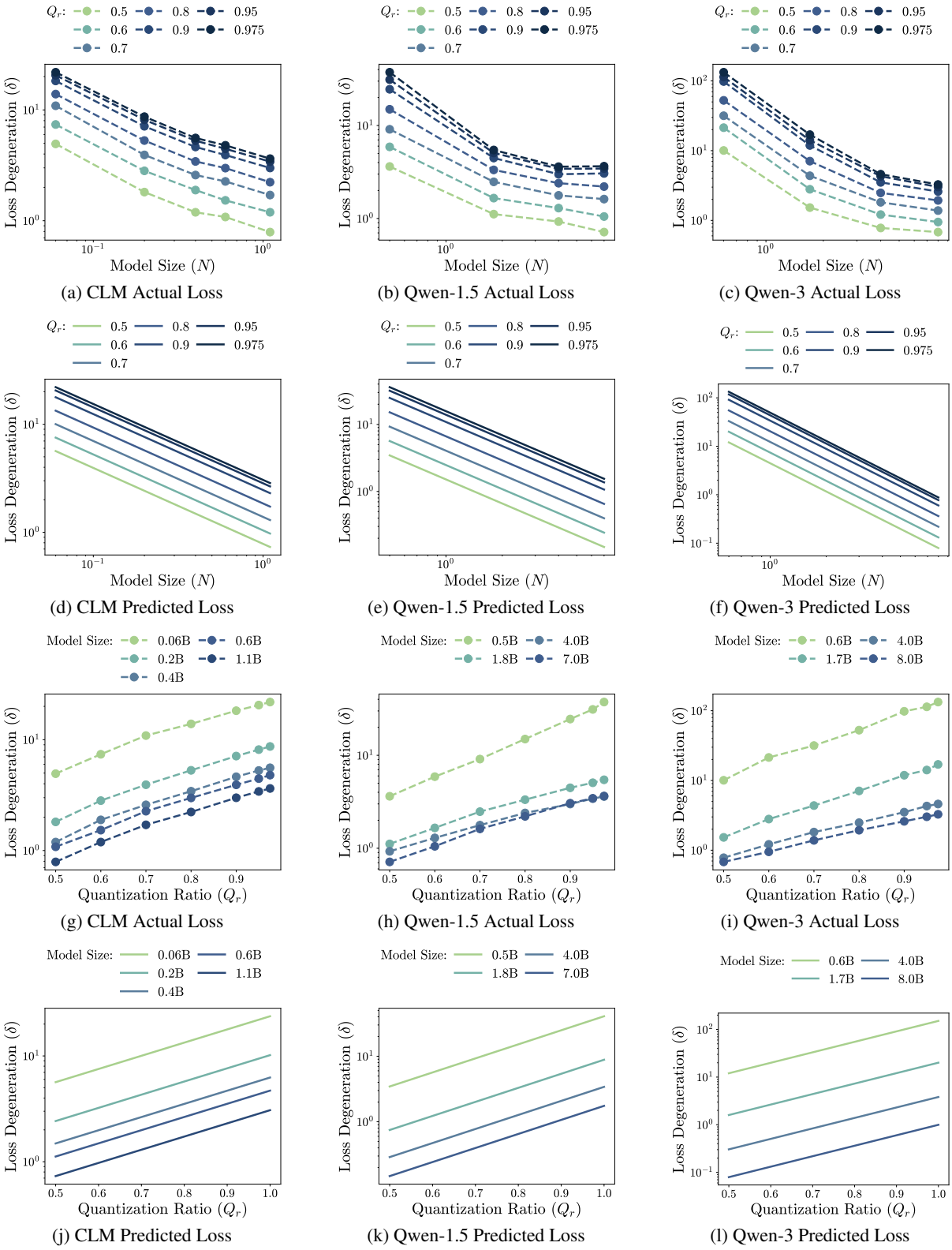
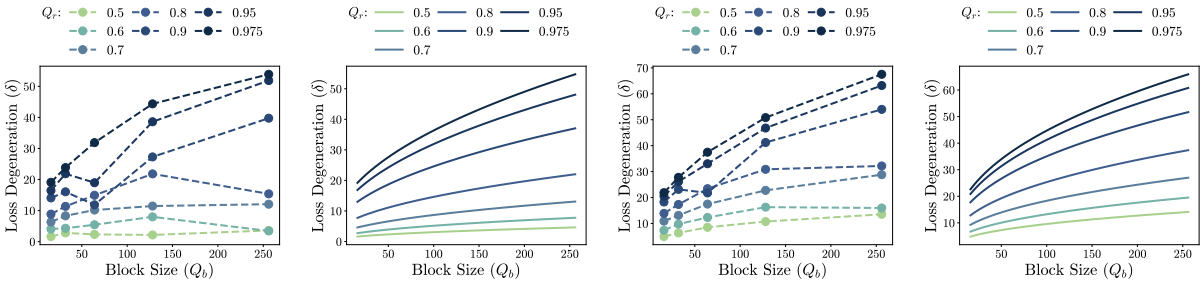
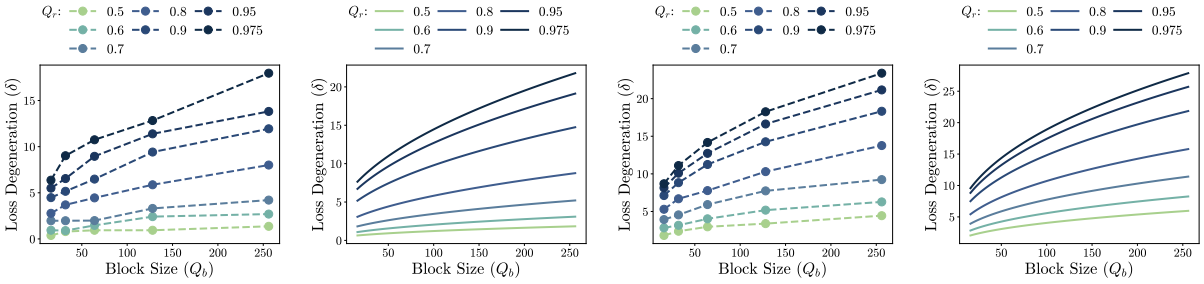


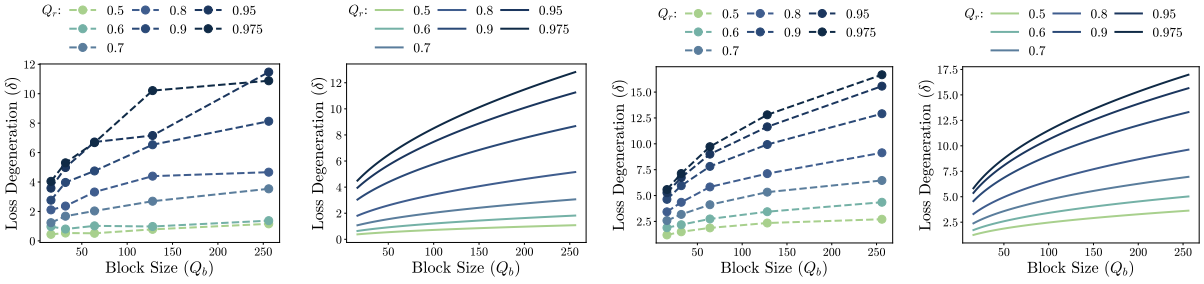

Figure 12: **Matrix Multiplication-wise ($\delta_\mu$)** (a,d,g,h) CLM matrix multiplication-wise results; (b,e,h,k) Qwen-1.5 matrix multiplication-wise results; (c,f,i,l) Qwen-3 matrix multiplication-wise results.



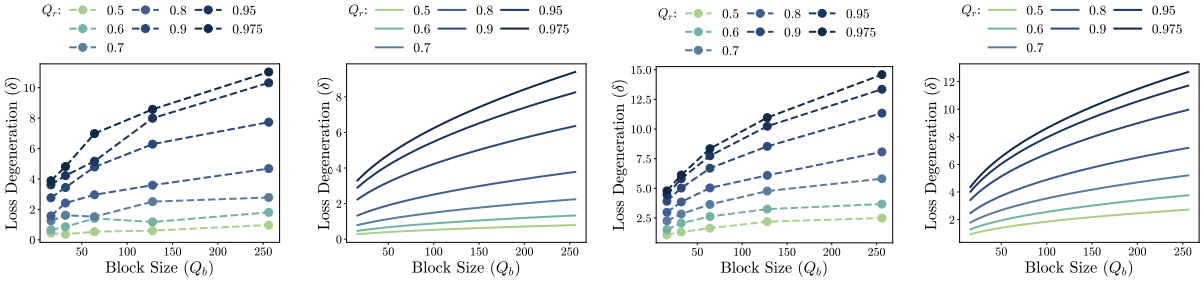

(a) 60M Actual (min) (b) 60M Predicted (min) (c) 60M Actual (mean) (d) 60M Predicted (mean)



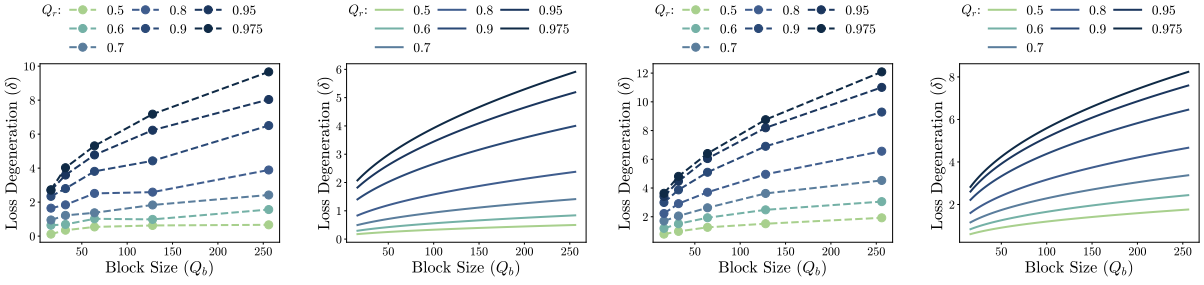

(e) 200M Actual (min) (f) 200M Predicted (min) (g) 200M Actual (mean) (h) 200M Predicted (mean)




(i) 400M Actual (min) (j) 400M Predicted (min) (k) 400M Actual (mean) (l) 400M Predicted (mean)

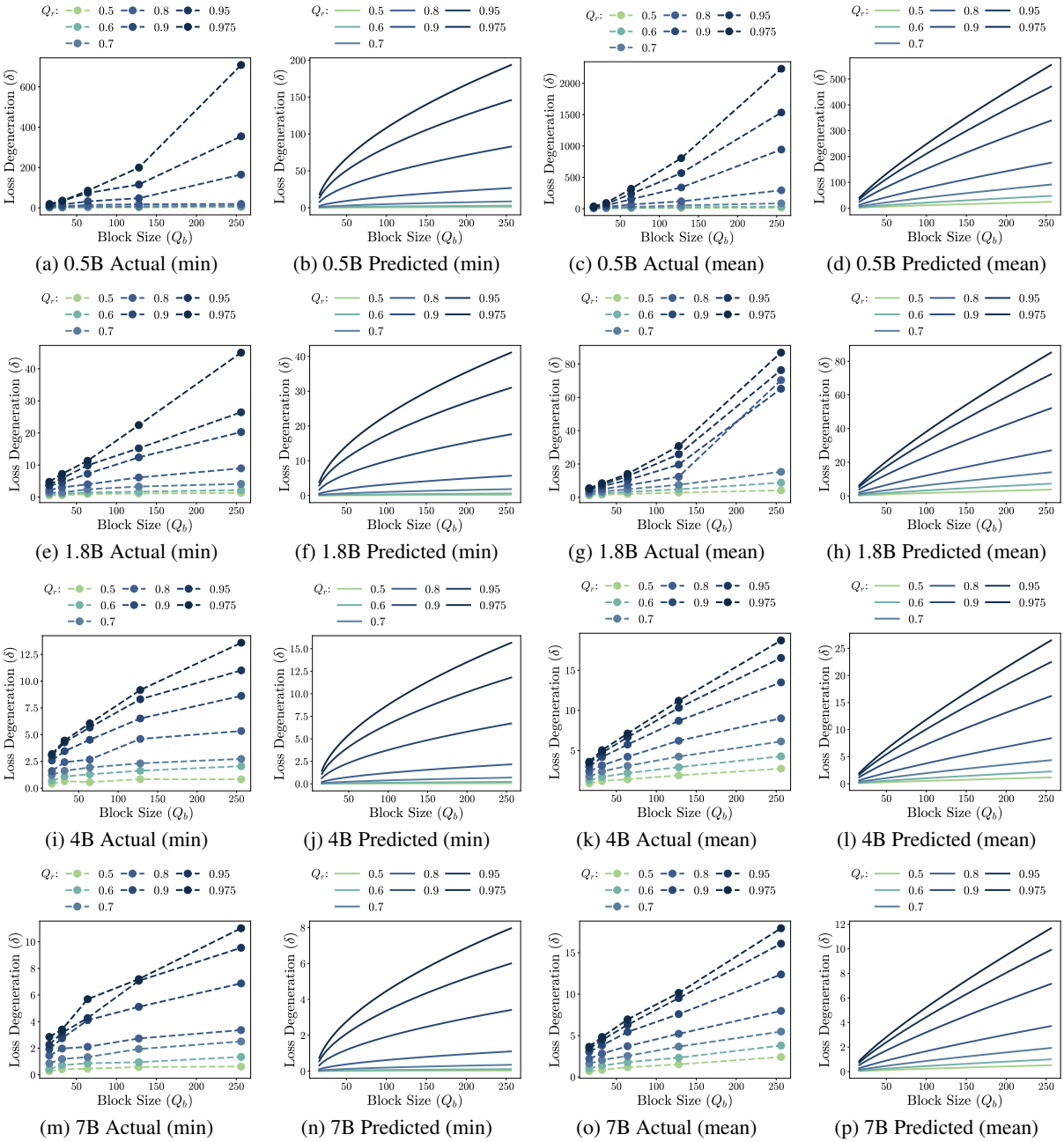



(m) 600M Actual (min) (n) 600M Predicted (min) (o) 600M Actual (mean) (p) 600M Predicted (mean)




(q) 1.1B Actual (min) (r) 1.1B Predicted (min) (s) 1.1B Actual (mean) (t) 1.1B Predicted (mean)

Figure 13: **Strong Law CLM Matrix Multiplication-wise ($\delta^{\text{opt}}$, $\delta_\mu$)** (a,b,e,f,i,j,m,n,q,r) CLM matrix multiplication-wise $\delta^{\text{opt}}$ results; (c,d,g,h,k,l,o,p,s,t) CLM matrix multiplication-wise $\delta_\mu$ results.



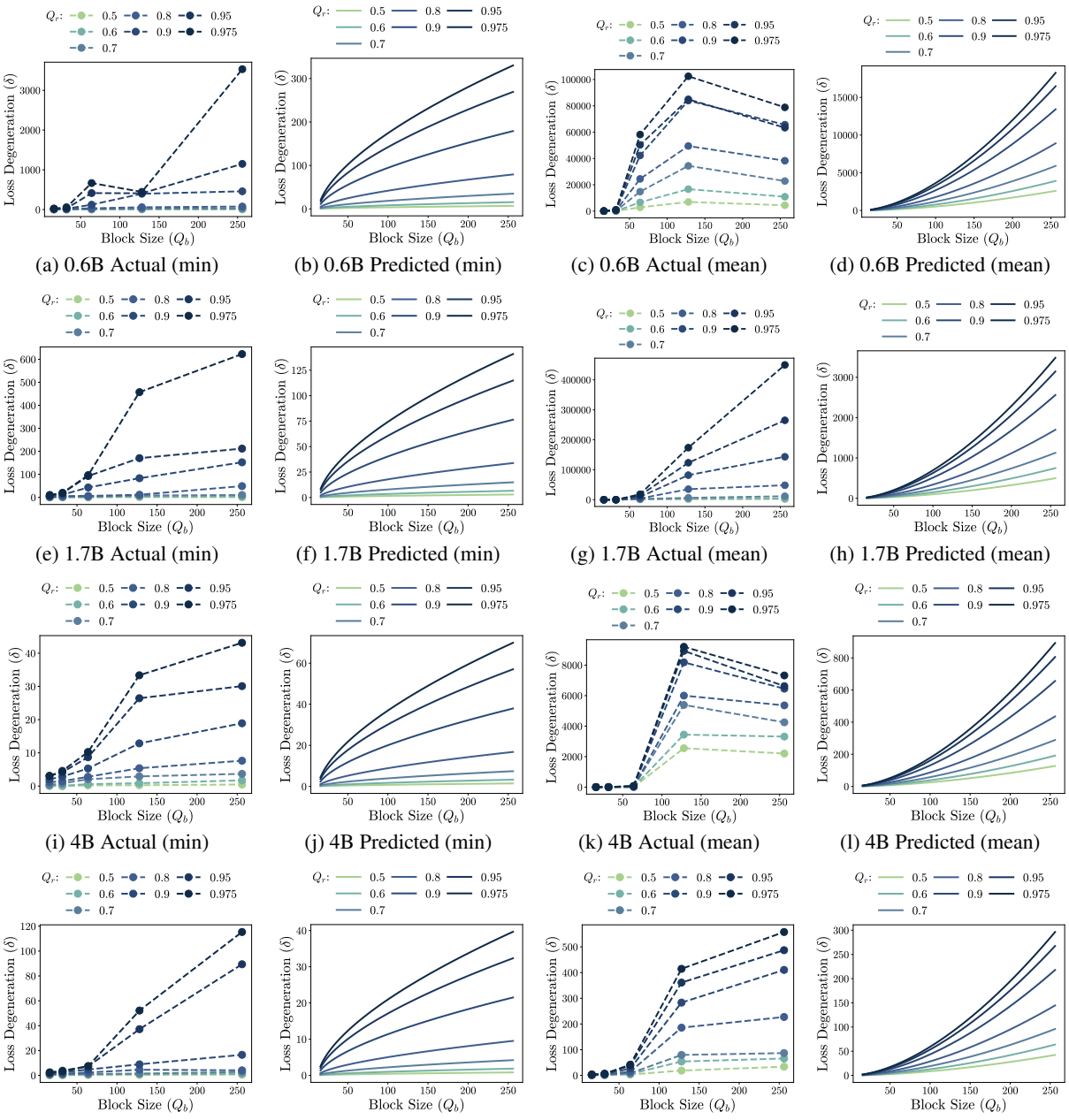

(a) 0.5B Actual (min) (b) 0.5B Predicted (min) (c) 0.5B Actual (mean) (d) 0.5B Predicted (mean)

(e) 1.8B Actual (min) (f) 1.8B Predicted (min) (g) 1.8B Actual (mean) (h) 1.8B Predicted (mean)

(i) 4B Actual (min) (j) 4B Predicted (min) (k) 4B Actual (mean) (l) 4B Predicted (mean)

(m) 7B Actual (min) (n) 7B Predicted (min) (o) 7B Actual (mean) (p) 7B Predicted (mean)

Figure 14: **Strong Law Qwen-1.5 Matrix Multiplication-wise ($\delta^{\text{opt}}$, $\delta_\mu$)** (a,b,e,f,i,j,m,n) Qwen1.5 matrix multiplication-wise $\delta^{\text{opt}}$ results; (c,d,g,h,k,l,o,p) Qwen-1.5 matrix multiplication-wise $\delta_\mu$ results.




(a) 0.6B Actual (min) (b) 0.6B Predicted (min) (c) 0.6B Actual (mean) (d) 0.6B Predicted (mean)

(e) 1.7B Actual (min) (f) 1.7B Predicted (min) (g) 1.7B Actual (mean) (h) 1.7B Predicted (mean)

(i) 4B Actual (min) (j) 4B Predicted (min) (k) 4B Actual (mean) (l) 4B Predicted (mean)

(m) 8B Actual (min) (n) 8B Predicted (min) (o) 8B Actual (mean) (p) 8B Predicted (mean)

Figure 15: **Strong Law Qwen-3 Matrix Multiplication-wise ($\delta^{\text{opt}}$, $\delta_\mu$)** (a,b,e,f,i,j,m,n) Qwen-3 matrix multiplication-wise $\delta^{\text{opt}}$ results; (c,d,g,h,k,l,o,p) Qwen-3 matrix multiplication-wise $\delta_\mu$ results.

even under high-performance quantization methods.

Note that, under these settings, there are some significant outliers for Qwen-3 results, with negative PTQ losses, such negative PTQ losses might be caused by the fact that *all Qwen-3 models involved in our experiments are partially distilled from larger models* as indicated in (Yang et al., 2025), invalidating our assumptions in Sec. 2.2.

Figs. 18 and 19 show the results for layerwise GPTQ quantization. Figs. 20 and 21 show the results for Minifloat quantization. Figs. 22 and 23 show the results for MXFP quantization. The GPTQ quantization for the Qwen-3 architecture is infeasible since the architecture is not supported by the GPTQ method. Our fitted contours match the remaining CLM and Qwen-1.5 models.

In contrast, Figs. 24 and 25 show the results for MXINT-2 quantization, which performs poorly across the board. The loss increases significantly under this setting, suggesting that aggressive quantization in both weights and activations severely degrades model quality. Despite the high loss, our fitted contours remain aligned with the empirical observations, capturing the underlying loss surface accurately.

Figs. 26 and 27 display the loss contours for MXINT-4 weight-only quantization. This method shows moderate performance. It incurs a smaller degradation in loss than MXINT-2. The fitted contours once again align well with actual measurements, reinforcing the robustness of our scaling law across varying quantization granularities and levels of aggressiveness. Similar to the HQQ quantization, the effectiveness of MXINT-4 weight-only quantization and the distillation training process of Qwen-3 explains the outliers for Qwen-3 figures.

## F Formula Fitting

We report all fitted coefficients and $R^2$ plots in Figs. 28 to 53. These include block-wise strong law fittings, as well as weak law fittings under layerwise, matrix-multiplication-wise, HQQ, MXINT-2, and MXINT-4 weight-only quantization. Each table reports the coefficients described in Sec. 2.2 from our scaling law equations, and the scatter plots compare predicted and actual PTQ losses to evaluate the fitting quality.

**Overall Trends and Implications.** Across all quantization levels and model families, the $R^2$ plots validate the predictive power of our scaling laws. Tight clustering along the diagonal confirms that the fitted models not only approximate empirical losses well but also generalize across compression levels and model sizes. Exceptions, including negative $\gamma_N$ or inflated constants, indicate model-instability or ill-posed settings such as distillation pre-training or over-compression.

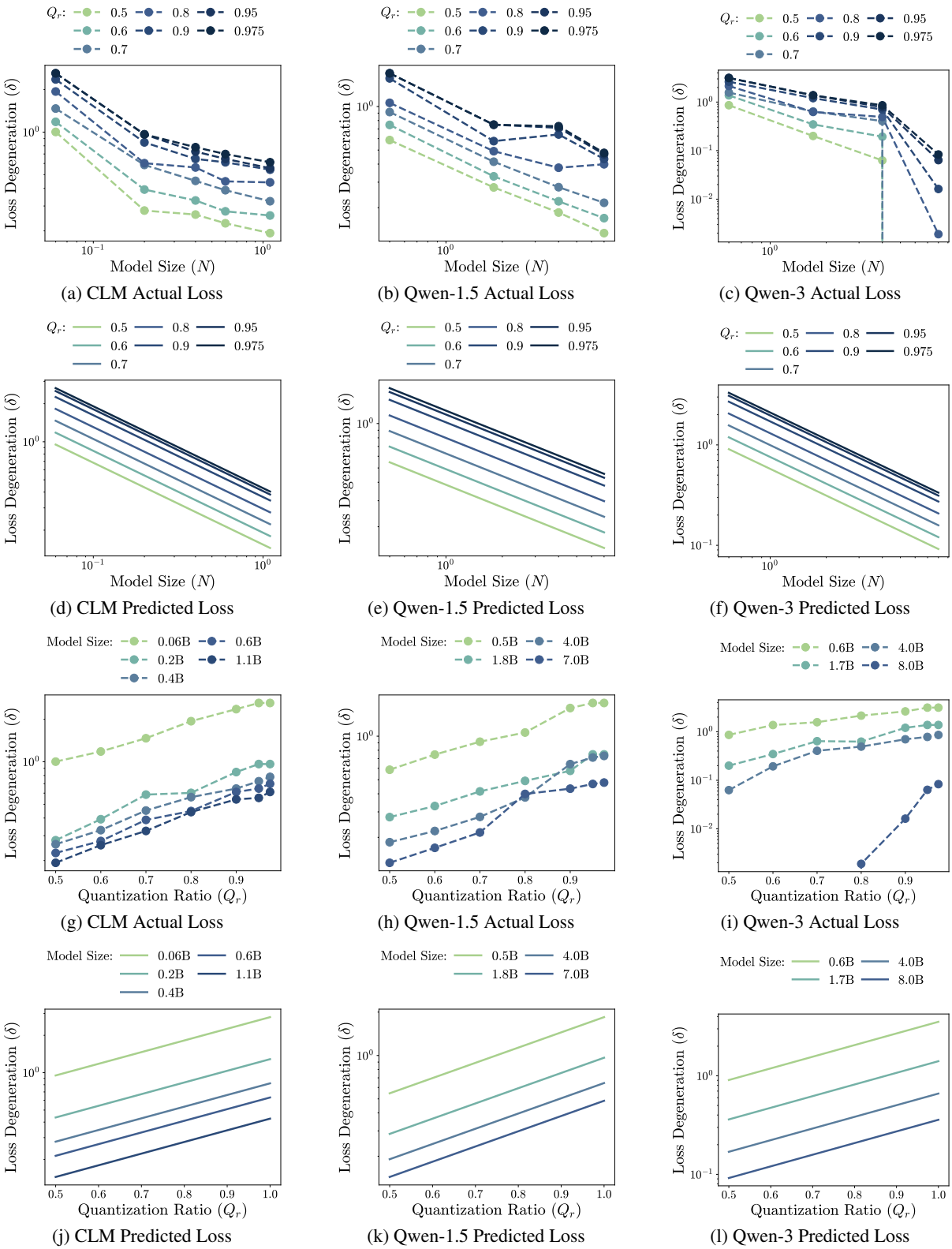
**Block-wise Strong Law Fitting** Figs. 28–30 display the strong law fits across model families. For CLM and Qwen-1.5, we observe high-quality fits with consistent scaling exponents. However, Qwen-3 shows greater variability, with large $\gamma_c$ values (e.g., 1.67), indicating strong sensitivity to the quantization ratio $Q_r$, likely caused by its distillation-only training pipeline.

**Layerwise and Matrix Multiplication-wise MXINT-4 Fitting** Figs. 45–50 show the layerwise and matrix multiplication-wise MXINT-4 quantization weak law results in strong, consistent fits across CLM and Qwen-1.5.

**HQQ, GPTQ, Minifloat, MXFP Fitting** Figs. 31–41 demonstrate that HQQ, GPTQ, Minifloat, and MXFP quantizations maintain consistently good fits across all model families. These weak law results further confirm the strength of those quantization methods in preserving model accuracy after quantization, and they validate that our scaling law generalizes even under high-performance quantizers.

**MXINT-2 Fitting** Figs. 42–44 illustrate that MXINT-2—our most aggressive quantization—yields degraded fits. This is especially evident for Qwen-1.5, where fitted constants reach extreme values (e.g., $C = 1181.1$), and for Qwen-3, where $\gamma_N$ becomes negative ($-4.62$), indicating that MXINT-2 quantization is not ideal. Despite this, the $R^2$ plots still show approximate alignment, suggesting that the weak law scaling structure is partially preserved.

**MXINT-4 Weight-only Fitting** Figs. 51-53 confirm that weight-only MXINT-4 quantization achieves a balanced tradeoff: it reduces parameter precision with less impact on performance compared to MXINT-2. Fit quality remains strong with reasonable coefficients. These trends validate that restricting quantization to weights alone preserves much of the original loss landscape structure, and our weak law accurately tracks this behavior across CLM, Qwen-1.5, and Qwen-3.




(a) CLM Actual Loss (b) Qwen-1.5 Actual Loss (c) Qwen-3 Actual Loss

(d) CLM Predicted Loss (e) Qwen-1.5 Predicted Loss (f) Qwen-3 Predicted Loss

(g) CLM Actual Loss (h) Qwen-1.5 Actual Loss (i) Qwen-3 Actual Loss

(j) CLM Predicted Loss (k) Qwen-1.5 Predicted Loss (l) Qwen-3 Predicted Loss

Figure 16: **HQQ** ($\delta^{\text{opt}}$) (a,d,g,h) CLM HQQ results; (b,e,h,k) Qwen-1.5 HQQ results; (c,f,i,l) Qwen-3 HQQ results.

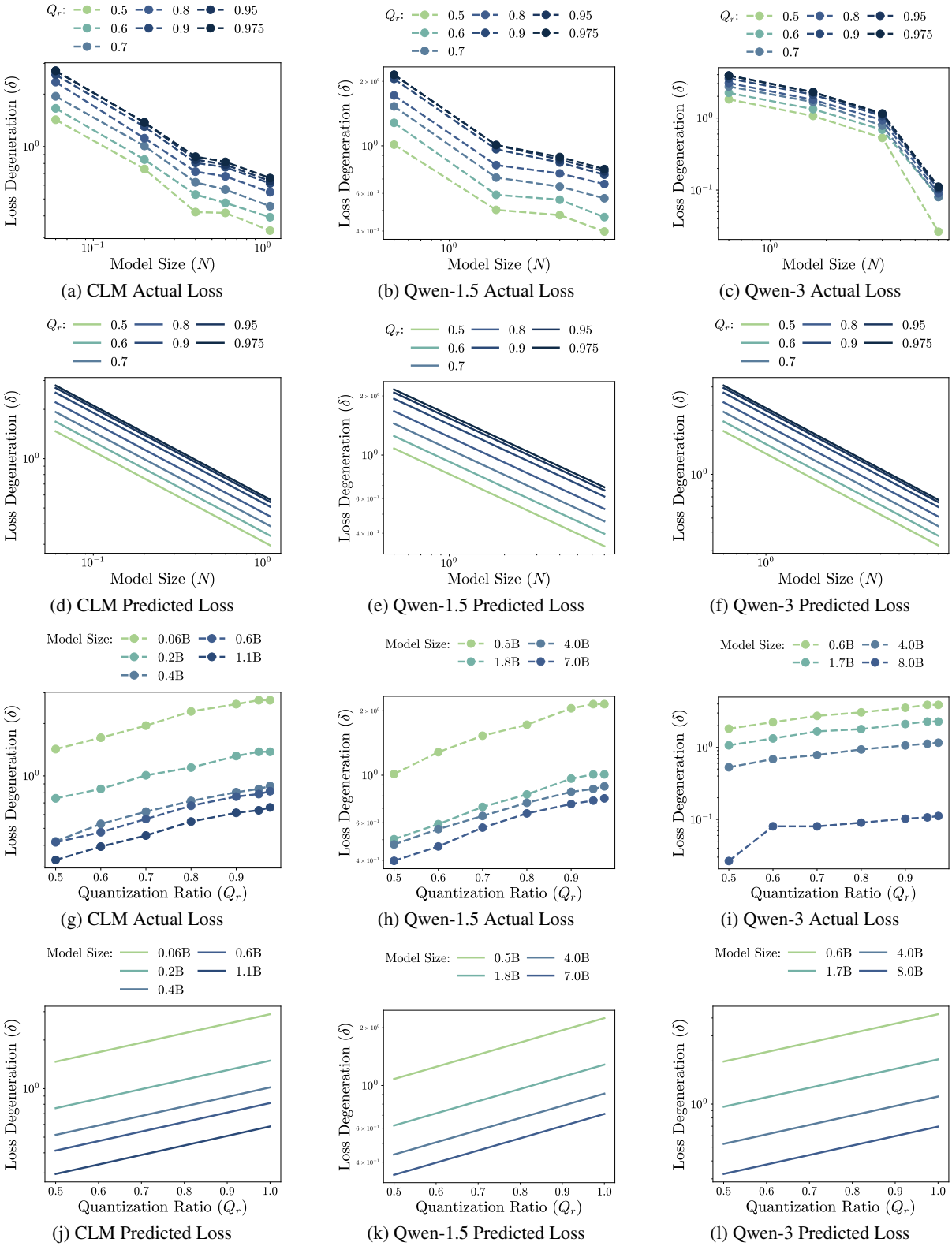
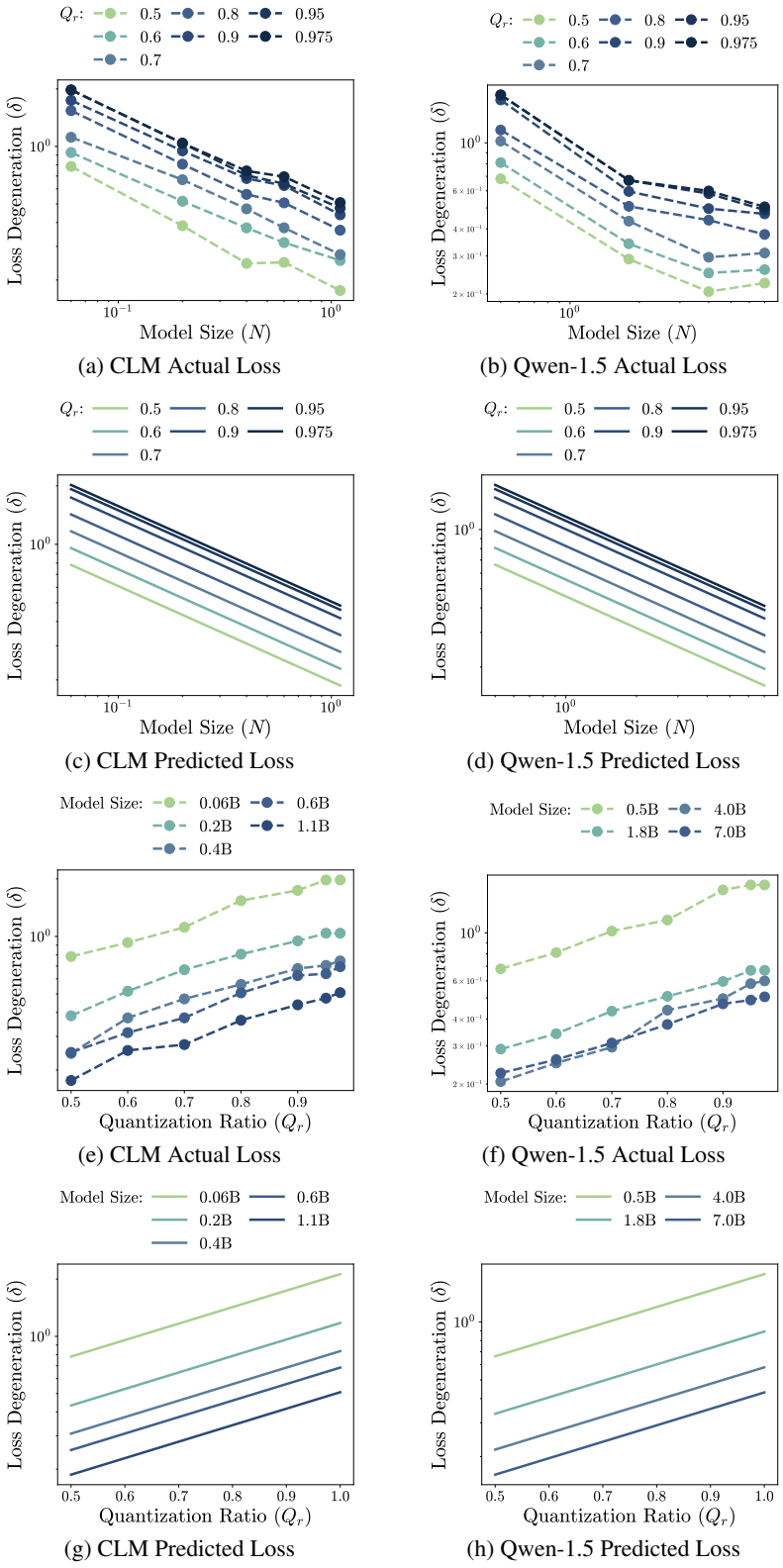



Figure 17: **HQQ** ($\delta_\mu$) (a,d,g,h) CLM HQQ results; (b,e,h,k) Qwen-1.5 HQQ results; (c,f,i,l) Qwen-3 HQQ results.



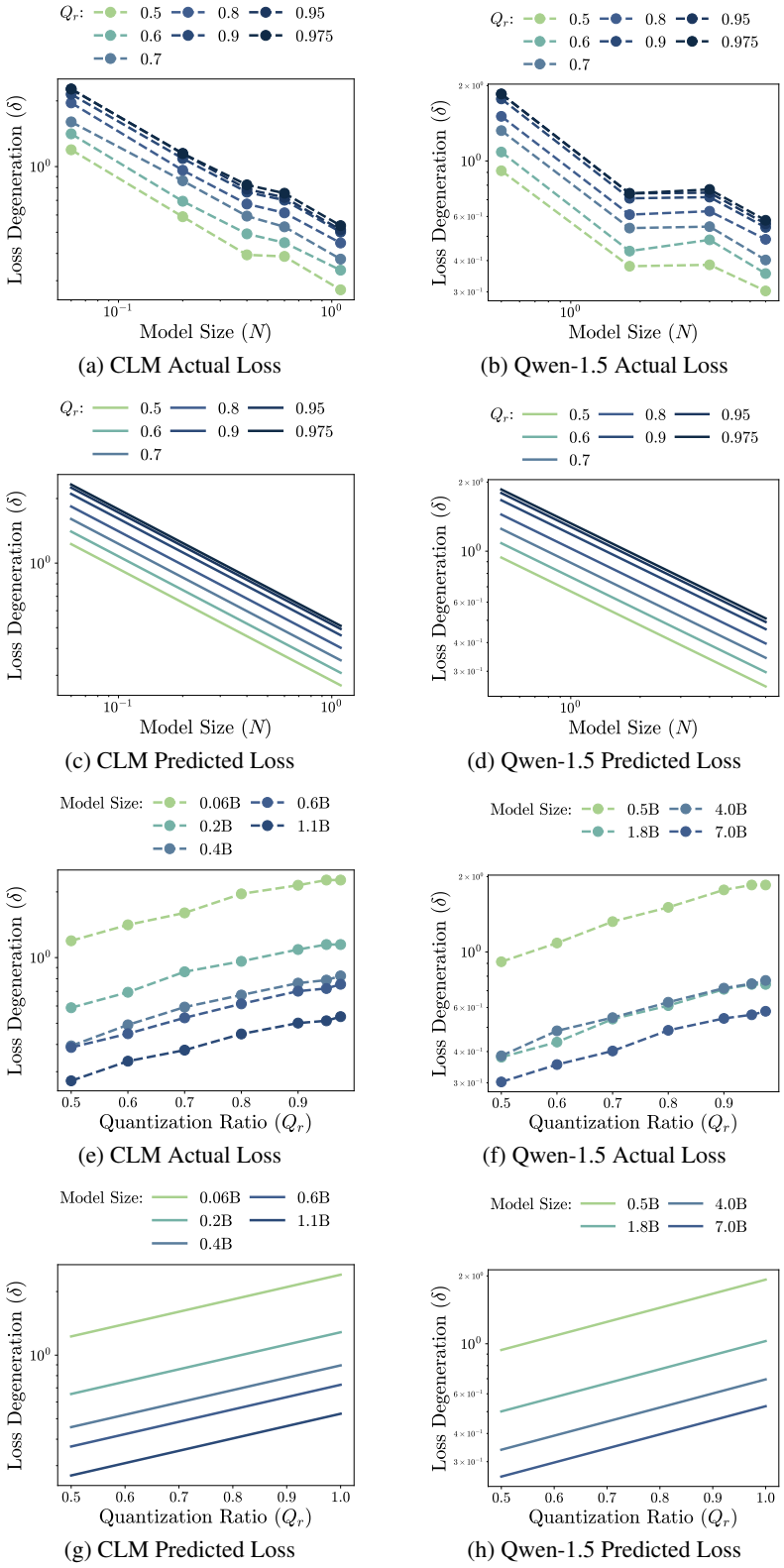

(a) CLM Actual Loss (b) Qwen-1.5 Actual Loss

(c) CLM Predicted Loss (d) Qwen-1.5 Predicted Loss

(e) CLM Actual Loss (f) Qwen-1.5 Actual Loss

(g) CLM Predicted Loss (h) Qwen-1.5 Predicted Loss

Figure 18: **GPTQ** ($\delta^{\text{opt}}$) (a,c,e,g) CLM GPTQ results; (b,d,f,h) Qwen-1.5 GPTQ results.




(a) CLM Actual Loss (b) Qwen-1.5 Actual Loss

(c) CLM Predicted Loss (d) Qwen-1.5 Predicted Loss

(e) CLM Actual Loss (f) Qwen-1.5 Actual Loss

(g) CLM Predicted Loss (h) Qwen-1.5 Predicted Loss

Figure 19: **GPTQ** ($\delta_\mu$) (a,c,e,g) CLM GPTQ results; (b,d,f,h) Qwen-1.5 GPTQ results.

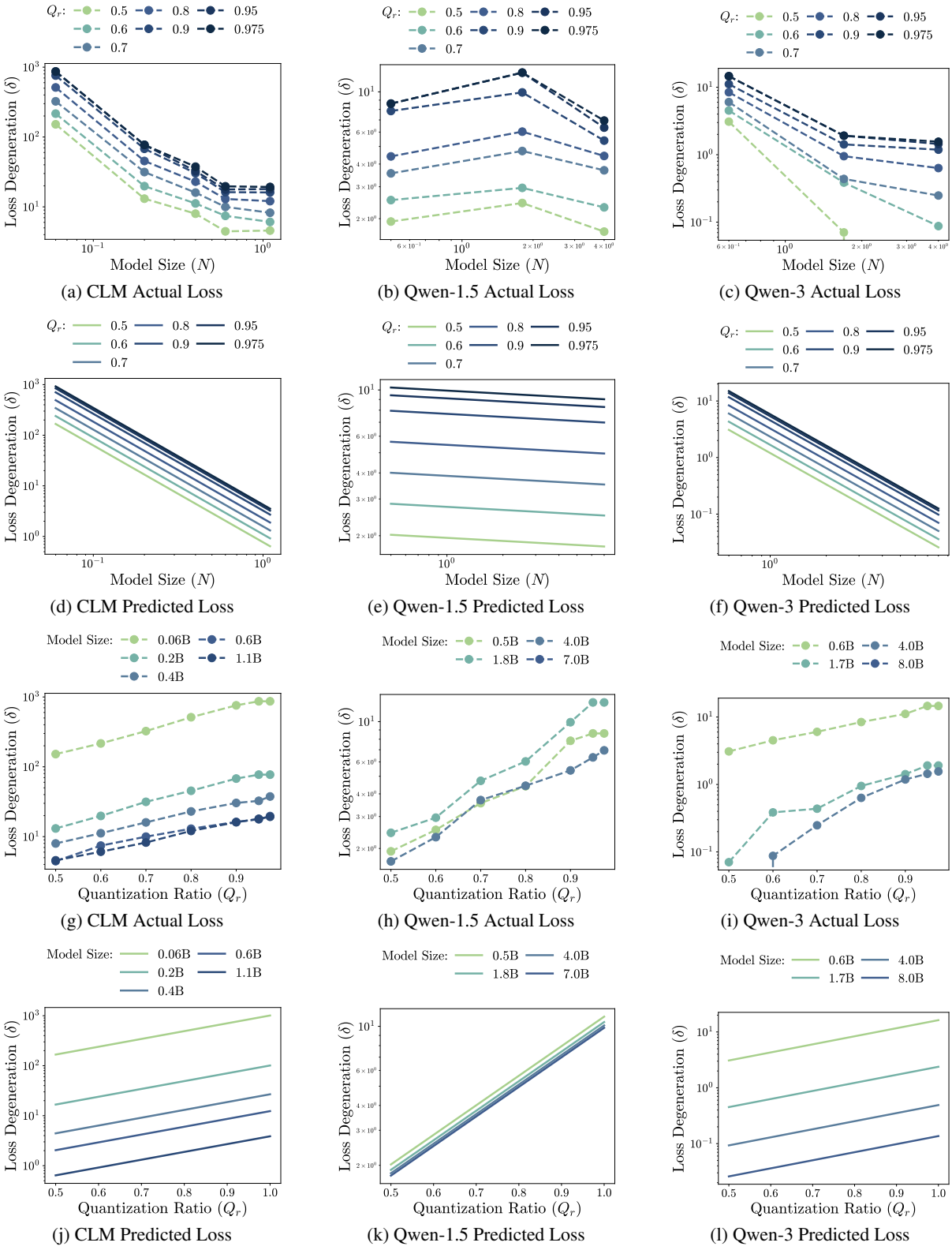



Figure 20: **Minifloat ($\delta^{\text{opt}}$)** (a,d,g,h) CLM Minifloat results; (b,e,h,k) Qwen-1.5 Minifloat results; (c,f,i,l) Qwen-3 Minifloat results.

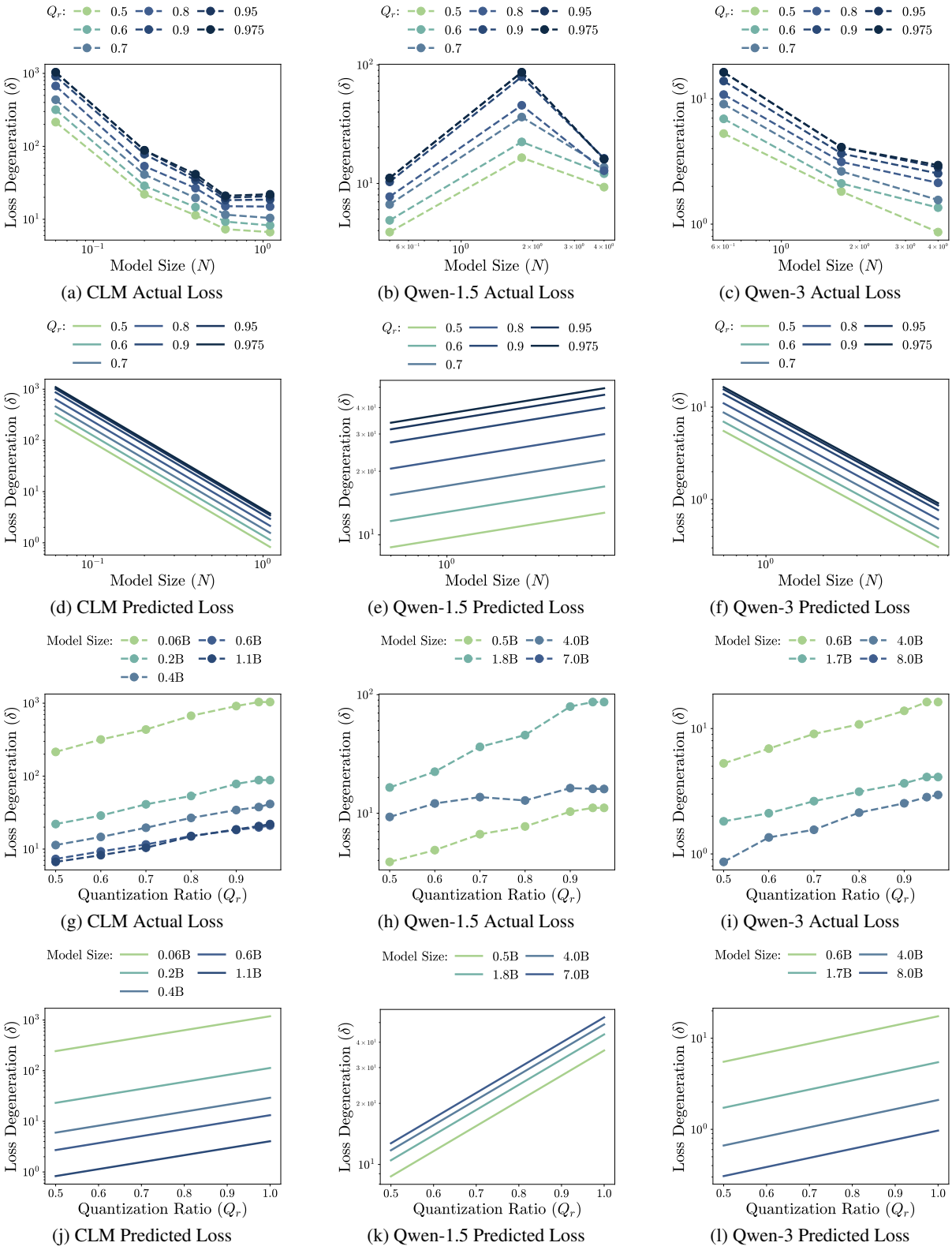
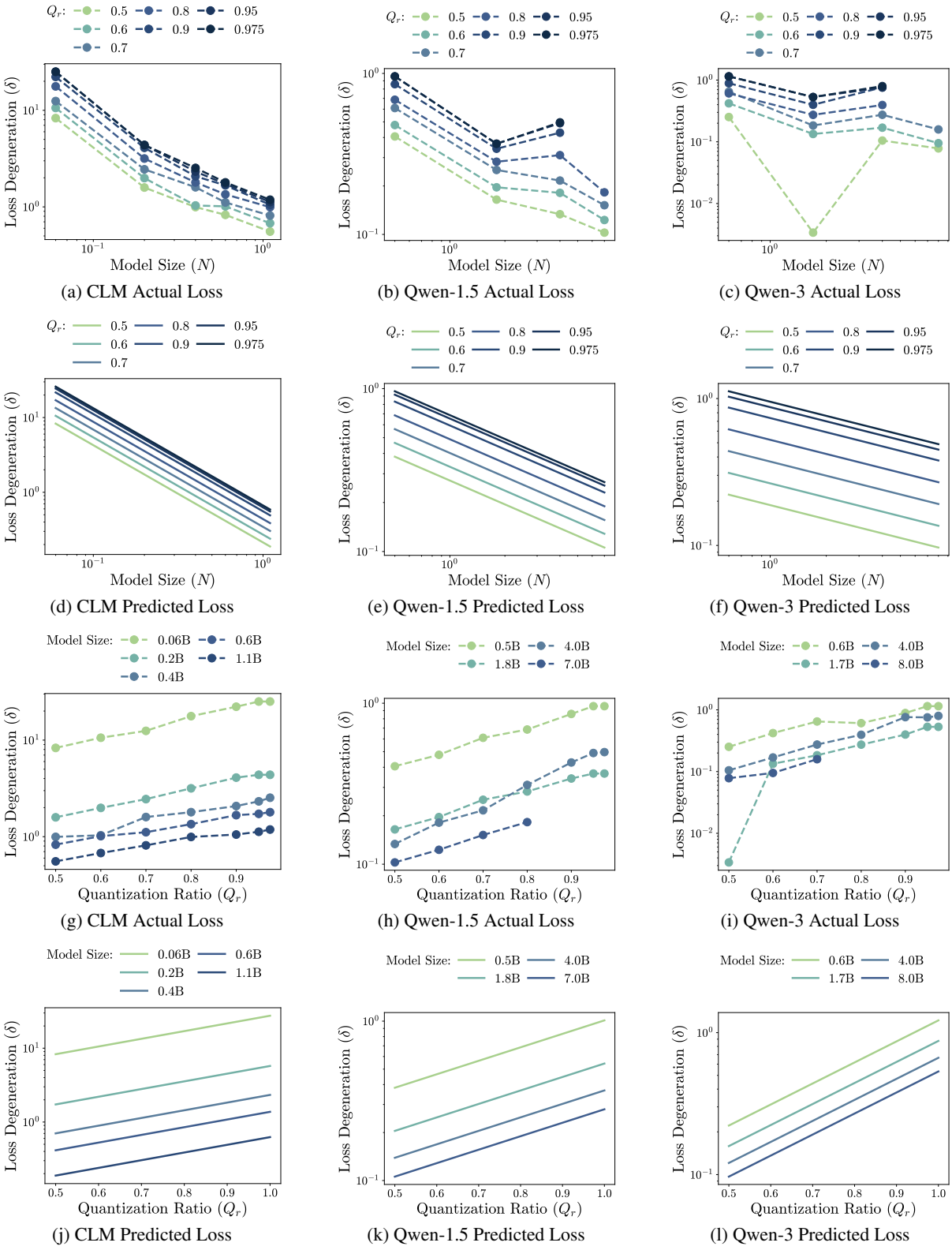



Figure 21: **Minifloat ($\delta_\mu$)** (a,d,g,h) CLM Minifloat results; (b,e,h,k) Qwen-1.5 Minifloat results; (c,f,i,l) Qwen-3 Minifloat results.



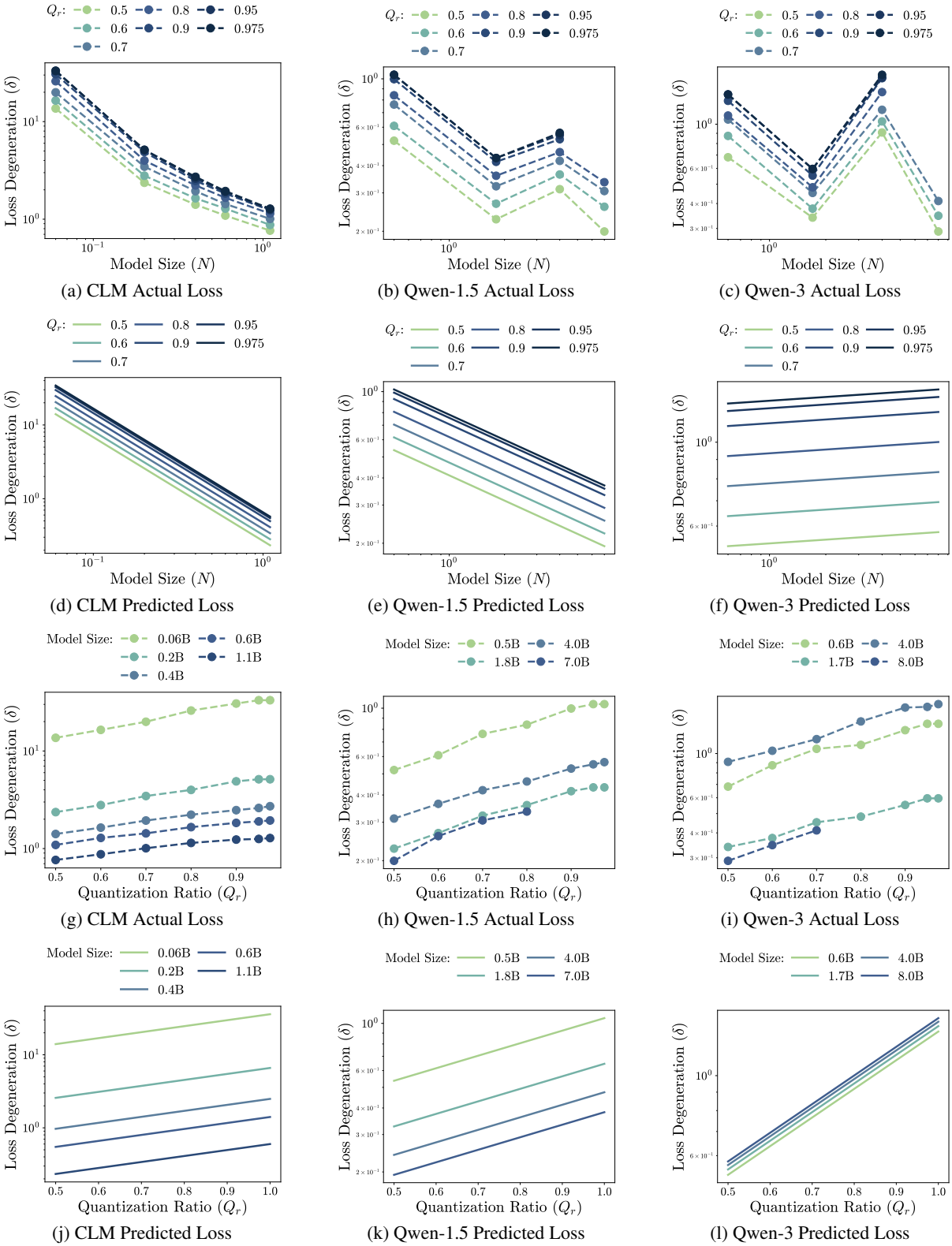

(a) CLM Actual Loss (b) Qwen-1.5 Actual Loss (c) Qwen-3 Actual Loss

(d) CLM Predicted Loss (e) Qwen-1.5 Predicted Loss (f) Qwen-3 Predicted Loss

(g) CLM Actual Loss (h) Qwen-1.5 Actual Loss (i) Qwen-3 Actual Loss

(j) CLM Predicted Loss (k) Qwen-1.5 Predicted Loss (l) Qwen-3 Predicted Loss

Figure 22: **MXFP** ($\delta^{\text{opt}}$) (a,d,g,h) CLM MXFP results; (b,e,h,k) Qwen-1.5 MXFP results; (c,f,i,l) Qwen-3 MXFP results.




(a) CLM Actual Loss (b) Qwen-1.5 Actual Loss (c) Qwen-3 Actual Loss

(d) CLM Predicted Loss (e) Qwen-1.5 Predicted Loss (f) Qwen-3 Predicted Loss

(g) CLM Actual Loss (h) Qwen-1.5 Actual Loss (i) Qwen-3 Actual Loss

(j) CLM Predicted Loss (k) Qwen-1.5 Predicted Loss (l) Qwen-3 Predicted Loss

Figure 23: **MXFP** ($\delta_\mu$) (a,d,g,h) CLM MXFP results; (b,e,h,k) Qwen-1.5 MXFP results; (c,f,i,l) Qwen-3 MXFP results.

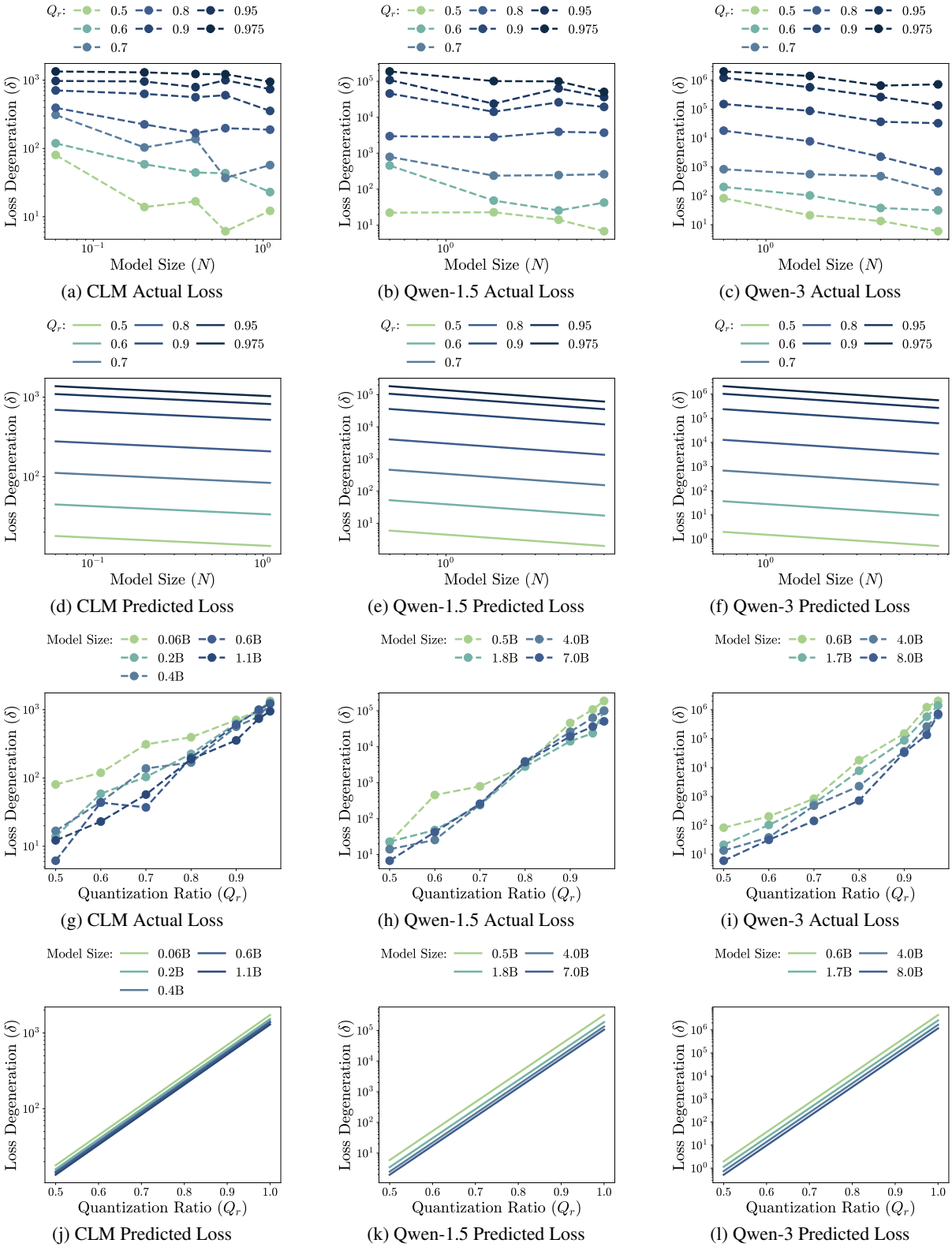



Figure 24: **MXINT-2 ($\delta^{\text{opt}}$)** (a,d,g,h) CLM MXINT-2 results; (b,e,h,k) Qwen-1.5 MXINT-2 results; (c,f,i,l) Qwen-3 MXINT-2 results.

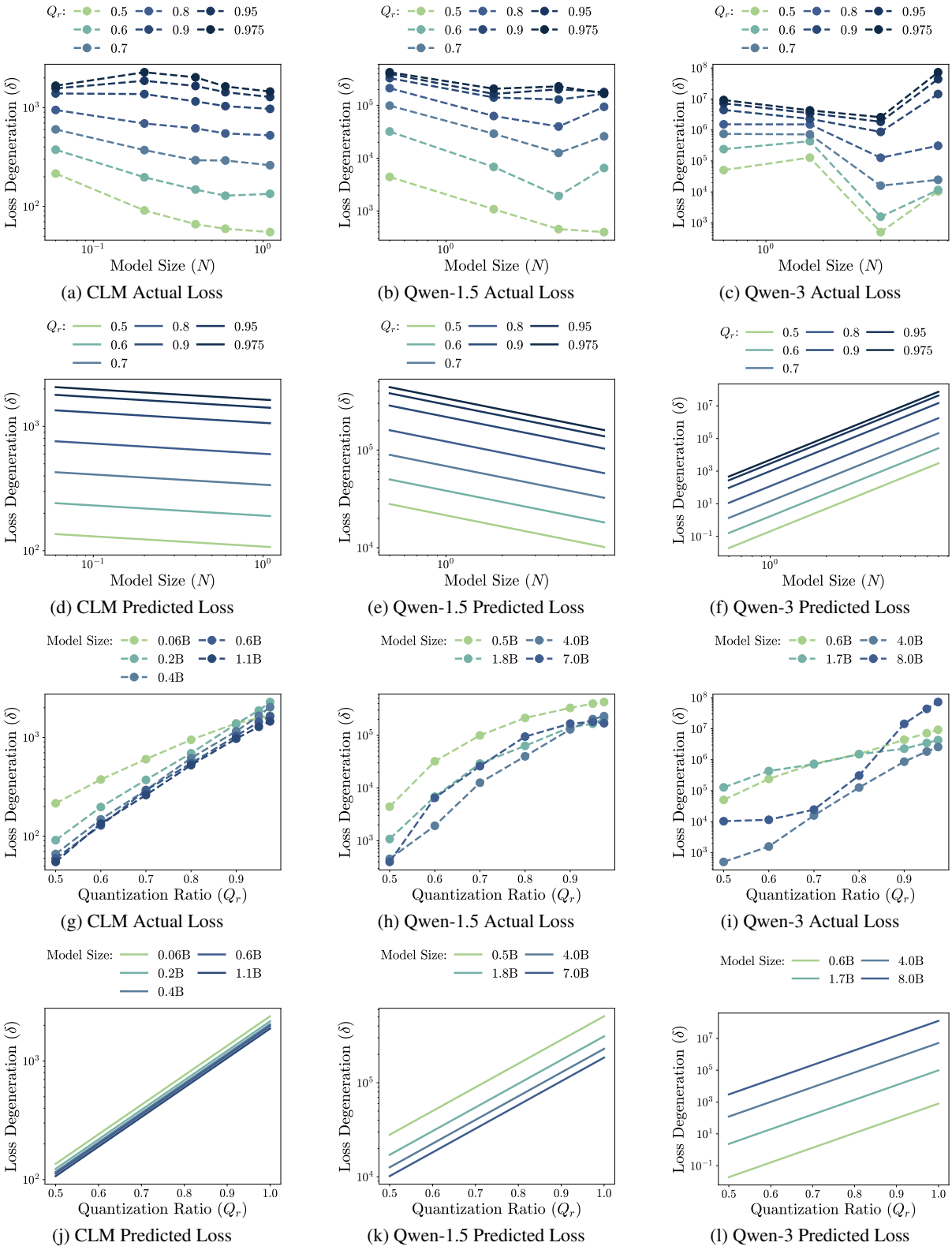
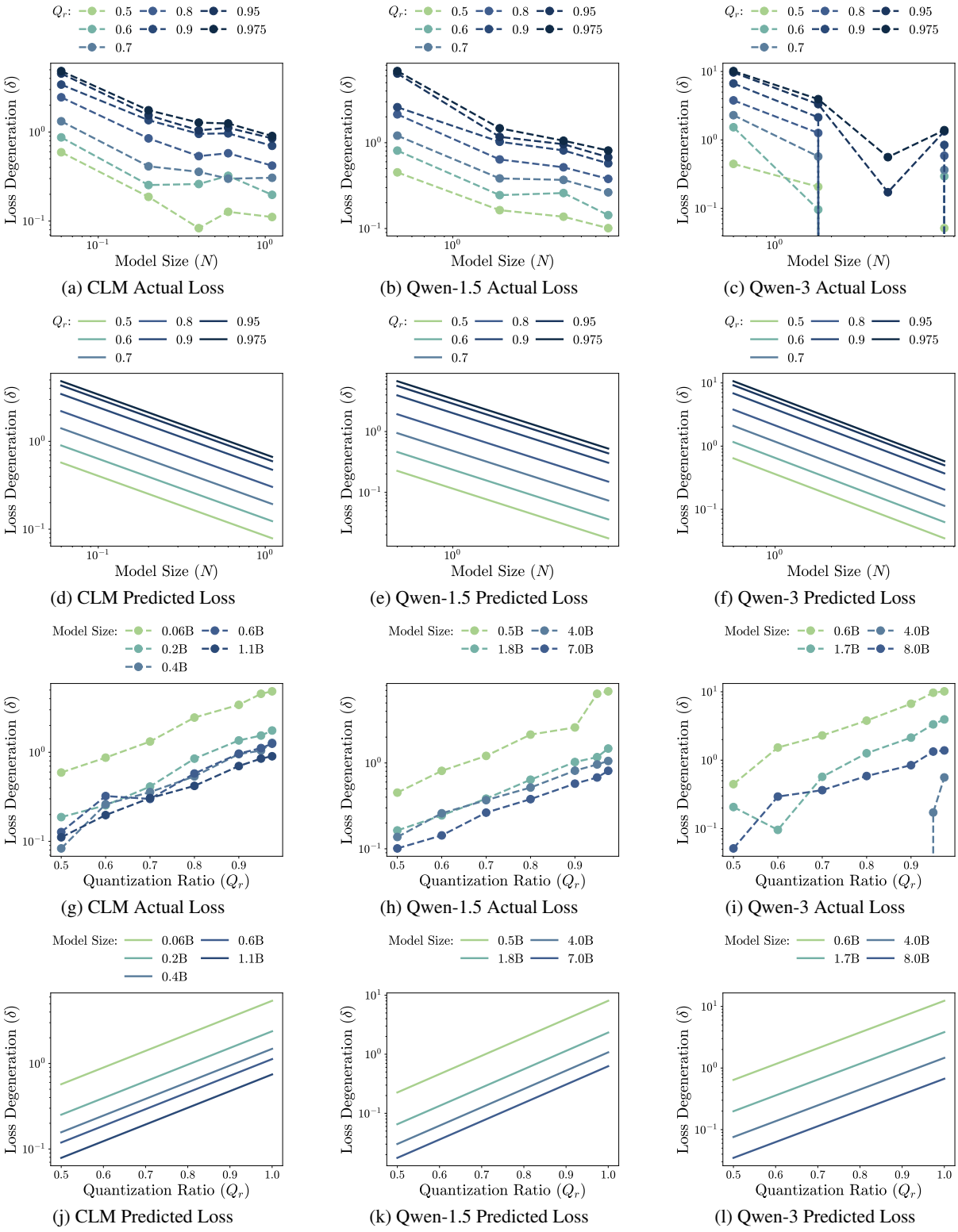



Figure 25: **MXINT-2** ($\delta_\mu$) (a,d,g,h) CLM MXINT-2 results; (b,e,h,k) Qwen-1.5 MXINT-2 results; (c,f,i,l) Qwen-3 MXINT-2 results.



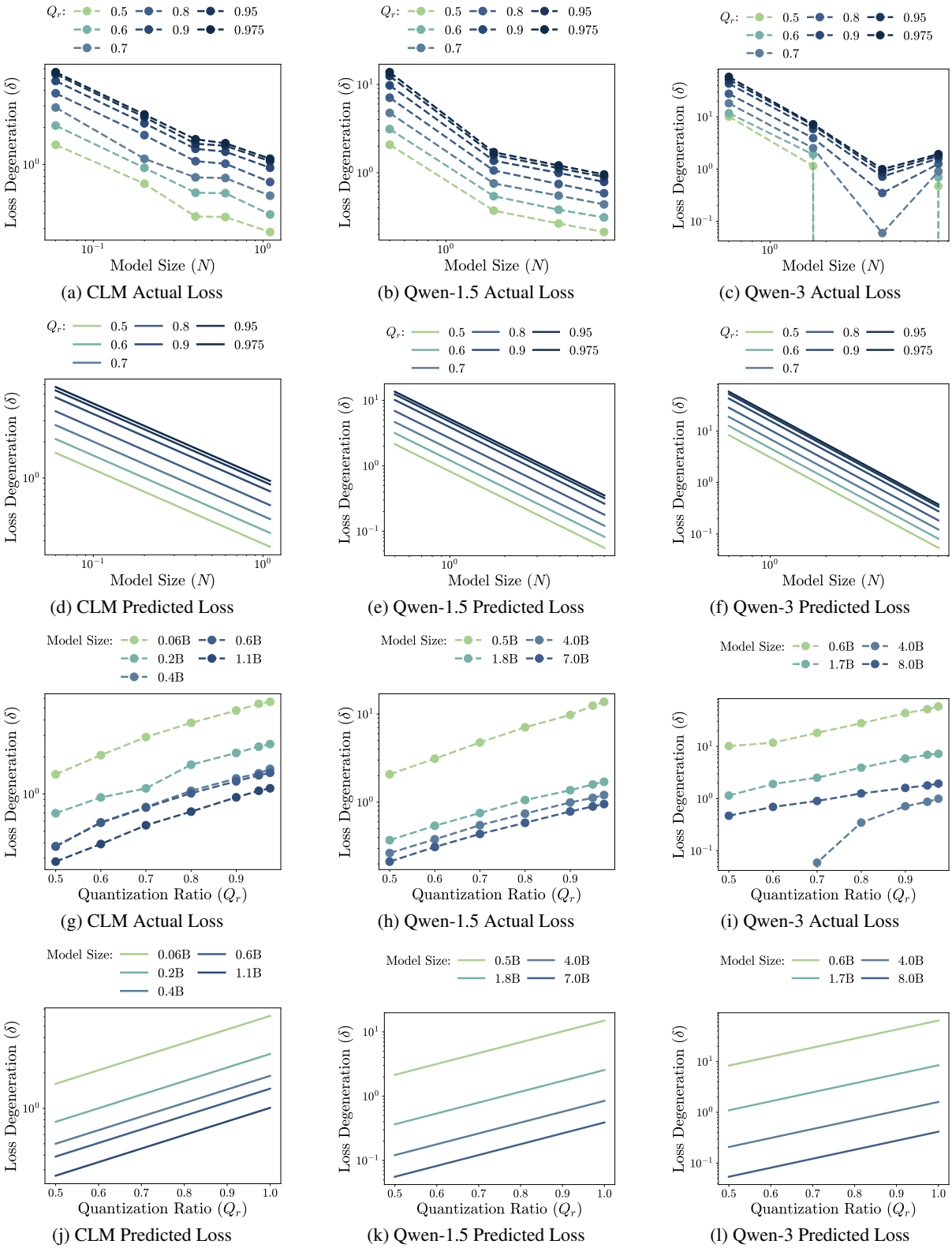

(a) CLM Actual Loss (b) Qwen-1.5 Actual Loss (c) Qwen-3 Actual Loss

(d) CLM Predicted Loss (e) Qwen-1.5 Predicted Loss (f) Qwen-3 Predicted Loss

(g) CLM Actual Loss (h) Qwen-1.5 Actual Loss (i) Qwen-3 Actual Loss

(j) CLM Predicted Loss (k) Qwen-1.5 Predicted Loss (l) Qwen-3 Predicted Loss

Figure 26: **MXINT-4 Weight-only** ($\delta^{\text{opt}}$) (a,d,g,h) CLM MXINT-4 Weight-only results; (b,e,h,k) Qwen-1.5 MXINT-4 Weight-only results; (c,f,i,l) Qwen-3 MXINT-4 Weight-only results.




(a) CLM Actual Loss (b) Qwen-1.5 Actual Loss (c) Qwen-3 Actual Loss

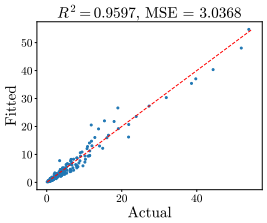
(d) CLM Predicted Loss (e) Qwen-1.5 Predicted Loss (f) Qwen-3 Predicted Loss

(g) CLM Actual Loss (h) Qwen-1.5 Actual Loss (i) Qwen-3 Actual Loss

(j) CLM Predicted Loss (k) Qwen-1.5 Predicted Loss (l) Qwen-3 Predicted Loss

Figure 27: **MXINT-4 Weight-only** ($\delta_\mu$) (a,d,g,h) CLM MXINT-4 Weight-only results; (b,e,h,k) Qwen-1.5 MXINT-4 Weight-only results; (c,f,i,l) Qwen-3 MXINT-4 Weight-only results.

| | |
|---|---|
| $C$ | 0.0028 |
| $A$ | 5.2055 |
| $\gamma_N$ | 0.7651 |
| $d$ | 13.632 |
| $\gamma_c$ | 0.4741 |




| | |
|---|---|
| $C$ | 0.0345 |
| $A$ | 3.2462 |
| $\gamma_N$ | 0.715 |
| $d$ | 5.4028 |
| $\gamma_c$ | 0.4273 |

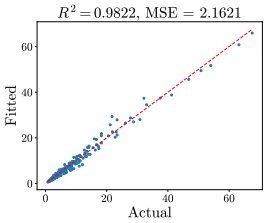
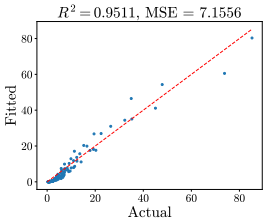



Figure 28: **CLM Strong Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.0001 |
| $A$ | 11.3016 |
| $\gamma_N$ | 1.2086 |
| $d$ | -12.2589 |
| $\gamma_c$ | 0.567 |




| | |
|---|---|
| $C$ | 0.0036 |
| $A$ | 6.557 |
| $\gamma_N$ | 1.4627 |
| $d$ | -5.4784 |
| $\gamma_c$ | 0.8237 |

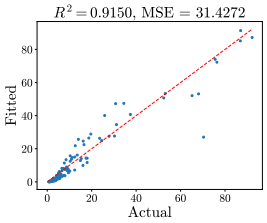
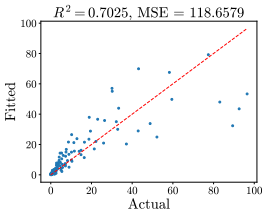



Figure 29: **Qwen-1.5 Strong Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.0025 |
| $A$ | 8.1566 |
| $\gamma_N$ | 0.8181 |
| $d$ | -13.1436 |
| $\gamma_c$ | 0.6208 |




| | |
|---|---|
| $C$ | 0.0142 |
| $A$ | 4.1025 |
| $\gamma_N$ | 1.5909 |
| $d$ | -3.7594 |
| $\gamma_c$ | 1.6736 |

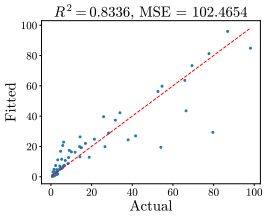
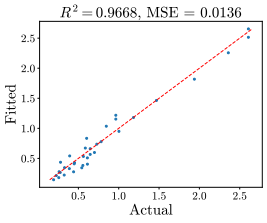



Figure 30: **Qwen-3 Strong Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.0522 |
| $A$ | 2.1631 |
| $\gamma_N$ | 0.6471 |




| | |
|---|---|
| $C$ | 0.1583 |
| $A$ | 1.3561 |
| $\gamma_N$ | 0.5508 |

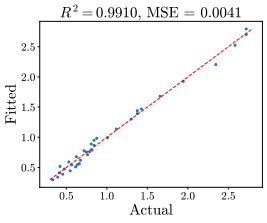
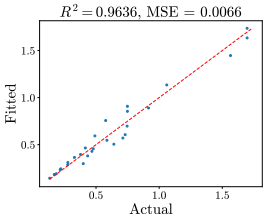



Figure 31: **HQQ CLM Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.114 |
| $A$ | 2.4336 |
| $\gamma_N$ | 0.5063 |




| | |
|---|---|
| $C$ | 0.3863 |
| $A$ | 1.4557 |
| $\gamma_N$ | 0.4341 |

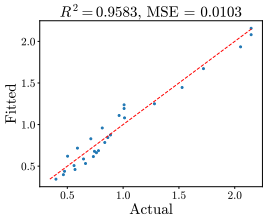
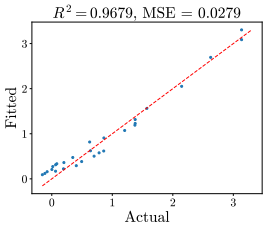



Figure 32: **HQQ Qwen-1.5 Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.148 |
| $A$ | 2.7216 |
| $\gamma_N$ | 0.8841 |




| | |
|---|---|
| $C$ | 0.6476 |
| $A$ | 1.5246 |
| $\gamma_N$ | 0.699 |

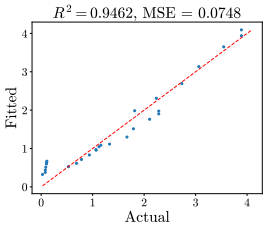
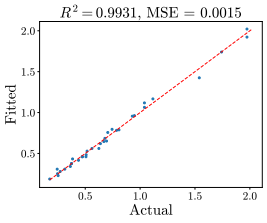



Figure 33: **HQQ Qwen-3 Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.0522 |
| $A$ | 2.1631 |
| $\gamma_N$ | 0.6471 |




| | |
|---|---|
| $C$ | 0.1583 |
| $A$ | 1.3561 |
| $\gamma_N$ | 0.5508 |

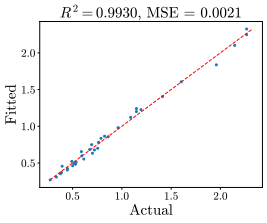
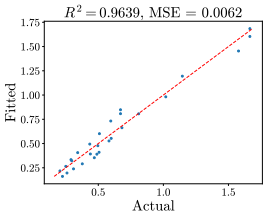



Figure 34: **GPTQ CLM Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.1140 |
| $A$ | 2.4336 |
| $\gamma_N$ | 0.5063 |




| | |
|---|---|
| $C$ | 0.3863 |
| $A$ | 1.4557 |
| $\gamma_N$ | 0.4341 |

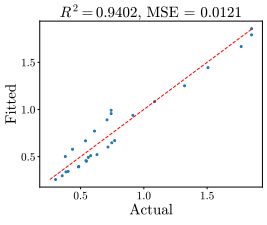
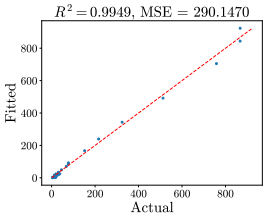



Figure 35: **GPTQ Qwen-1.5 Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.0522 |
| $A$ | 2.1631 |
| $\gamma_N$ | 0.6471 |




| | |
|---|---|
| $C$ | 0.1583 |
| $A$ | 1.3561 |
| $\gamma_N$ | 0.5508 |

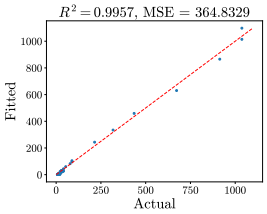
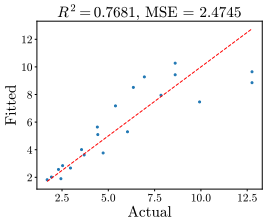



Figure 36: **Minifloat CLM Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.1140 |
| $A$ | 2.4336 |
| $\gamma_N$ | 0.5063 |




| | |
|---|---|
| $C$ | 0.3863 |
| $A$ | 1.4557 |
| $\gamma_N$ | 0.4341 |

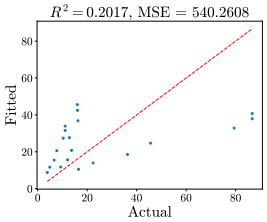
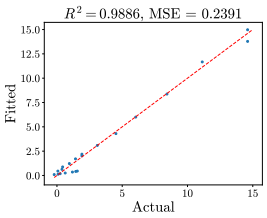



Figure 37: **Minifloat Qwen-1.5 Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.1480 |
| $A$ | 2.7216 |
| $\gamma_N$ | 0.8841 |




| | |
|---|---|
| $C$ | 0.6476 |
| $A$ | 1.5246 |
| $\gamma_N$ | 0.6990 |

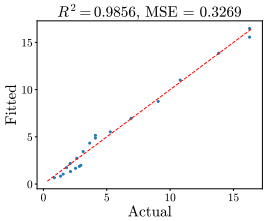
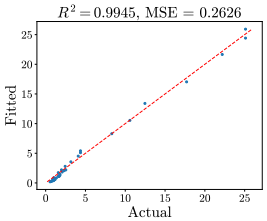



Figure 38: **Minifloat Qwen-3 Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.0522 |
| $A$ | 2.1631 |
| $\gamma_N$ | 0.6471 |




| | |
|---|---|
| $C$ | 0.1583 |
| $A$ | 1.3561 |
| $\gamma_N$ | 0.5508 |

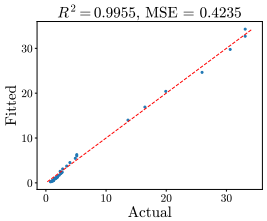
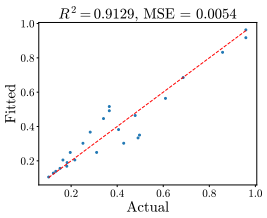



Figure 39: **MXFP CLM Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.1140 |
| $A$ | 2.4336 |
| $\gamma_N$ | 0.5063 |




| | |
|---|---|
| $C$ | 0.3863 |
| $A$ | 1.4557 |
| $\gamma_N$ | 0.4341 |

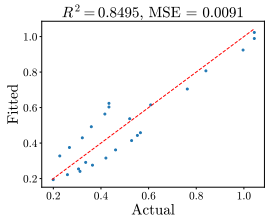
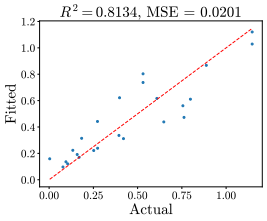



Figure 40: **MXFP Qwen-1.5 Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.1480 |
| $A$ | 2.7216 |
| $\gamma_N$ | 0.8841 |




| | |
|---|---|
| $C$ | 0.6476 |
| $A$ | 1.5246 |
| $\gamma_N$ | 0.6990 |

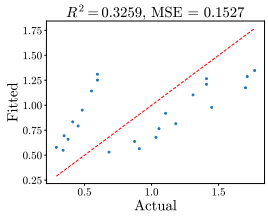
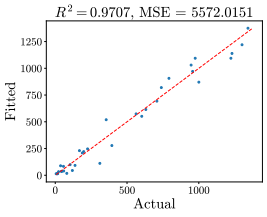



Figure 41: **MXFP Qwen-3 Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.1407 |
| $A$ | 9.1365 |
| $\gamma_N$ | 0.0992 |




| | |
|---|---|
| $C$ | 6.1421 |
| $A$ | 5.7318 |
| $\gamma_N$ | 0.0821 |

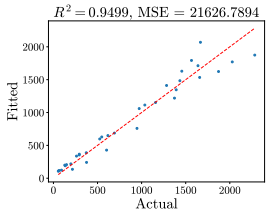
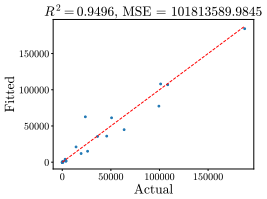



Figure 42: **MXINT-2 CLM Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.0001 |
| $A$ | 21.7819 |
| $\gamma_N$ | 0.4173 |




| | |
|---|---|
| $C$ | 1181.1394 |
| $A$ | 5.7997 |
| $\gamma_N$ | 0.3834 |

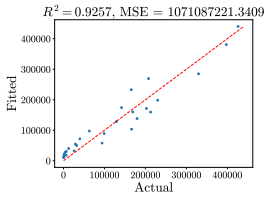
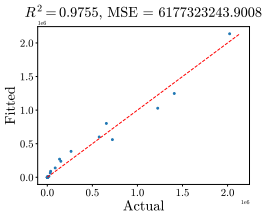



Figure 43: **MXINT-2 Qwen-1.5 Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.0 |
| $A$ | 29.2508 |
| $\gamma_N$ | 0.5165 |




| | |
|---|---|
| $C$ | 0.0 |
| $A$ | 21.2625 |
| $\gamma_N$ | -4.6183 |

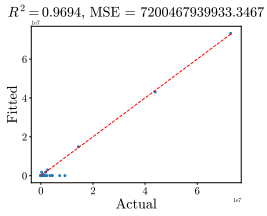
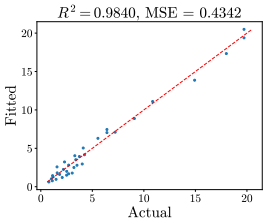



Figure 44: **MXINT-2 Qwen-3 Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.2187 |
| $A$ | 2.2312 |
| $\gamma_N$ | 0.8405 |




| | |
|---|---|
| $C$ | 0.6682 |
| $A$ | 1.465 |
| $\gamma_N$ | 0.7259 |

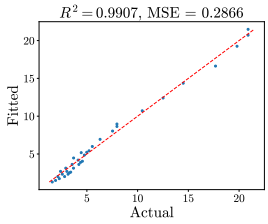
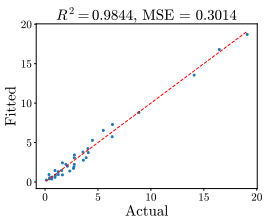



Figure 45: **MXINT-4 Layerwise CLM Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.0313 |
| $A$ | 4.2991 |
| $\gamma_N$ | 0.7828 |




| | |
|---|---|
| $C$ | 0.1862 |
| $A$ | 2.8693 |
| $\gamma_N$ | 0.703 |

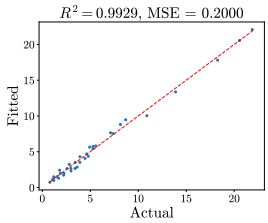
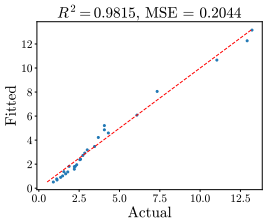



Figure 46: **MXINT-4 Matrix Multiplication-wise CLM Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$
.

| | |
|---|---|
| $C$ | 0.5173 |
| $A$ | 2.8049 |
| $\gamma_N$ | 0.7231 |




| | |
|---|---|
| $C$ | 1.0086 |
| $A$ | 2.2993 |
| $\gamma_N$ | 0.7686 |

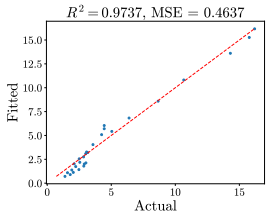
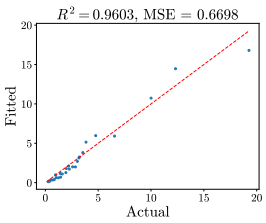



Figure 47: **MXINT-4 Layerwise Qwen-1.5 Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.0286 |
| $A$ | 5.9666 |
| $\gamma_N$ | 0.8073 |




| | |
|---|---|
| $C$ | 0.128 |
| $A$ | 4.9367 |
| $\gamma_N$ | 1.1931 |

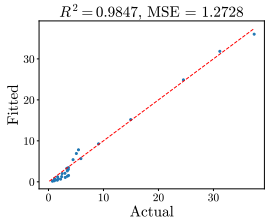
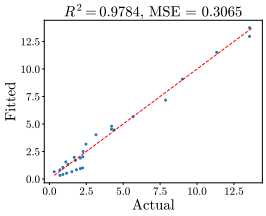



Figure 48: **MXINT-4 Matrix Multiplication-wise Qwen-1.5 Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.8134 |
| $A$ | 2.3706 |
| $\gamma_N$ | 1.0121 |




| | |
|---|---|
| $C$ | 1.5004 |
| $A$ | 1.9362 |
| $\gamma_N$ | 0.9633 |

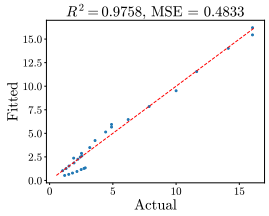
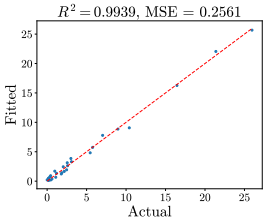



Figure 49: **MXINT-4 Layerwise Qwen-3 Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.0409 |
| $A$ | 6.0825 |
| $\gamma_N$ | 0.9994 |




| | |
|---|---|
| $C$ | 0.3562 |
| $A$ | 5.0609 |
| $\gamma_N$ | 1.9388 |

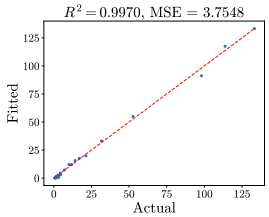
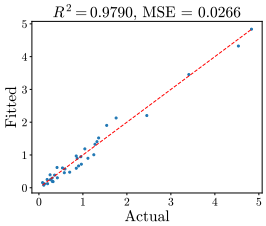



Figure 50: **MXINT-4 Matrix Multiplication-wise Qwen-3 Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.0089 |
| $A$ | 4.494 |
| $\gamma_N$ | 0.6818 |




| | |
|---|---|
| $C$ | 0.0749 |
| $A$ | 2.6608 |
| $\gamma_N$ | 0.6197 |

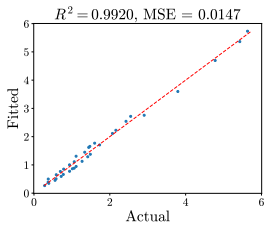
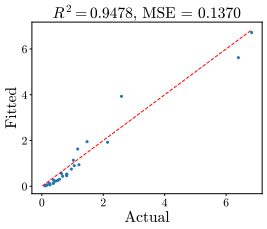



Figure 51: **Weight-only MXINT-4 CLM Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.0032 |
| $A$ | 7.1464 |
| $\gamma_N$ | 0.9663 |




| | |
|---|---|
| $C$ | 0.1179 |
| $A$ | 3.8872 |
| $\gamma_N$ | 1.3831 |

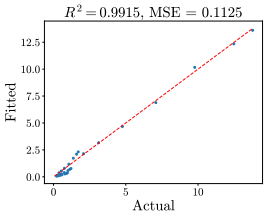
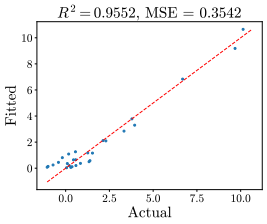



Figure 52: **Weight-only MXINT-4 Qwen-1.5 Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

| | |
|---|---|
| $C$ | 0.0188 |
| $A$ | 5.915 |
| $\gamma_N$ | 1.1251 |




| | |
|---|---|
| $C$ | 0.399 |
| $A$ | 4.0875 |
| $\gamma_N$ | 1.946 |

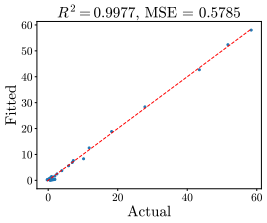



Figure 53: **Weight-only MXINT-4 Qwen-3 Weak Law.** The left two are for $\delta^{\text{opt}}$ and the right two are for $\delta_\mu$.

## G Further Explanation for our Estimation of $\mathbb{E}(\Delta)$ and $\min(\Delta)$

As discussed in Sec. 2.2, we are interested in two parameters of the discrete random variable $\Delta$: $\mathbb{E}(\Delta)$ and $\min(\Delta)$, which represent the mean and minimum (optimal) values of $\Delta$. In the weak law, they are denoted as $\delta^{\text{opt}}(N, Q_r)$ and $\mathbb{E}[\Delta(N, Q_r)]$. In the strong law, they are denoted as $\delta^{\text{opt}}(N, Q_r, Q_b)$ and $\mathbb{E}[\Delta(N, Q_r, Q_b)]$.

We estimate those two parameters with the following two estimators. Given a list of i.i.d. discrete random variables $\{\Delta_i\}_{i=1}^n$. We denote the unbiased estimator of $\mathbb{E}(\Delta)$ as

$$E_n = \frac{\sum_{i=1}^n \Delta_i}{n}, \tag{12}$$

and the biased estimator of $\min(\Delta)$ as

$$M_n = \min(\{\Delta_i : i = 1, \cdots, n\}). \tag{13}$$

$E_n$ is an unbiased estimator as

$$\begin{aligned} \mathbb{E}(E_n) - \mathbb{E}(\Delta) &= \frac{\sum_{i=1}^n \mathbb{E}(\Delta_i)}{n} - \mathbb{E}(\Delta) \\ &= \mathbb{E}(\Delta) - \mathbb{E}(\Delta) \\ &= 0. \end{aligned} \tag{14}$$

On the other hand, $M_n$ is a biased estimator. Without loss of generality, we will calculate the bias of $M'_n = \max(\{\Delta_i : i = 1, \cdots, n\})$ with respect to $\max(\Delta)$. Since $\{\Delta_i\}_{i=1}^n$ are i.i.d,

$$\begin{aligned} \Pr(M'_n < x) &= \prod_{i=1}^n \Pr(\Delta_i < x) \\ &= F^n(x^-), \end{aligned} \tag{15}$$

where $F$ is the cdf of $\Delta$ and $x^- = \max(\{x' : x' < x \wedge \Pr(\Delta = x') \neq 0\} \cup -\infty)$. Hence the expectation of $M'_n$ is

$$\begin{aligned} \mathbb{E}(M'_n) &= \sum_x x \Pr(M'_n = x) \\ &= \sum_x x \Pr(M'_n \leq x \wedge M'_n \not< x) \\ &= \sum_x x(F^n(x) - F^n(x^-)). \end{aligned} \tag{16}$$

As

$$\begin{aligned} &\sum_x (F^n(x) - F^n(x^-)) \\ &= F^n(\max(\Delta)) - F^n(-\infty) = 1 - 0 = 1, \end{aligned} \tag{17}$$

$\mathbb{E}(M'_n)$ is a re-weighting of different $x$-s and the weight for $\max(\Delta)$ is not 1, hence $\mathbb{E}(M'_n) < \max(\Delta)$. However, as $n \to \infty$, $(F^n(x) -$

$F^n(x^-)) \to 0$ for all $x \neq \max(\Delta)$, so

$$\mathbb{E}(M'_n) \xrightarrow{d} \max(\Delta) \text{ as } n \to \infty\,, \tag{18}$$

making $\{M'_n\}_{i=1}^{\infty}$ asymptotically unbiased. Recall notations from Sec. 2.2, given $n$ realizations $\{\delta_i\}_{i=1}^n$ of the random variables $\{\Delta_i\}_{i=1}^n$,

$$\mu_\delta = \frac{\sum_{i=1}^n \delta_i}{n} \tag{19}$$

and

$$\delta_{\min} = \min(\{\delta_i : i = 1, \cdots, n\}) \tag{20}$$

are our final estimation of the mean and minimum of $\Delta$. Our weak and strong law are designed to fit those estimations. To be specific, each fitting of the weak or the strong law tries to fit separately on $\mu_\delta$ and $\delta_{\min}$, which is also why each of our laws has two separate parts for expectation and minimum values.

## H Validation at Larger Model Scales

Table 4 reports fits of our scaling laws at larger model scales (13B, 14B, and 32B) under MXINT-4 quantization. We additionally ran layer-wise and matrix-multiplication-wise mixed quantization on Qwen-1.5-14B, Qwen-3-14B, LLaMA-13B, Qwen-1.5-32B, and Qwen-3-32B, using the same evaluation protocol described in Sec. 4.1 (subsampled SlimPajama perplexity, $Q_r \in \{0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.975\}$, 100 random partitions per configuration).

Across all settings, the fitted-vs-actual $R^2$ remains $\geq 0.96$ for both $\delta^{\text{opt}}$ and $\mathbb{E}[\delta]$, and the MSE stays within the same range as the smaller-scale fits reported in Sec. 4. We did not refit per-model constants; we evaluated the law fitted on $N \leq 8$B on these larger checkpoints and computed goodness-of-fit. This is a stronger form of validation than fitting on the same set: predicted losses at 13–32B are obtained purely by extrapolation along $N$, and the high $R^2$ values indicate that our scaling law extrapolates to the largest open-source dense LLMs we can run within our compute budget.

## I Additional Related Work

In addition to the backgrounds presented in Sec. 5, we further discuss the related work on mixed precision. Specifically, we categorize the work under the following three topics: mixed-precision quantization, mixed-precision inference, and mixed-precision training.

**Mixed-precision quantization** Mixed precision approaches involve partitioning a model's parameters into both high-precision and low-precision components, which have been shown to better preserve model performance relative to uniform quantization. This is primarily seen in models that exhibit different sensitivities to quantization at various layers. Some mixed-precision LLM quantization work adopts the concept of weight salience to guide the search for fine-grained bit allocation. The first-order (Li et al., 2023) or second-order weight gradient (Huang et al., 2024) has been used to form such salience metrics, such that salient layers are left in higher precision while the rest are cast to low precision. There are also works performing the search in an end-to-end style with the quantized model performance as the objective, such as the accuracy on a downstream task (Zhang et al., 2023). In both cases, mixed precision can be seen as a promising approach to provide a lossless reduction in LLM memory requirements, reducing average bit widths below levels achievable through uniform quantization.

**Mixed-precision inference** Mixed-precision inference methods targeting GPUs usually adopt regular mixed-precision strategies and computation patterns; The authors of GPTQ3.int8() (Dettmers et al., 2022) decompose the matrix multiplication in every linear layer into two sub-matrix-multiplications based on the activation magnitudes, achieving a 2-3× inference speedup by casting the low-magnitude sub-matrix to low precision. SpQR (Dettmers et al., 2023) represents a weight matrix with grouped 3-bit integers and less than 1% sensitive weight elements with FP16 values, achieving a 2× speedup compared to a quantized and sparse PyTorch baseline. These approaches enable reducing model size, but additional careful treatment is needed to improve inference throughput. For example, in (Li et al., 2023), mixed precision LLM quantization at 2-bit and 3-bit showed no speedup compared to 4-bit, due to less efficient utilization of memory bandwidth. On the other hand, Any-Precision LLM (Park et al., 2024) achieves throughput scaling at various precisions by providing CUDA kernels with a novel weight packing approach following a bitplane layout, achieving 1.3-1.8× speedup on mobile and edge devices. Additionally, works such as FlightLLM achieve high throughput by leveraging custom hardware designs (Zeng et al., 2024).

Table 4: Validation of the proposed scaling law at larger model scales under MXINT-4. Each row reports goodness-of-fit of our scaling law's prediction against the measured PTQ loss across $Q_r \in \{0.5, \ldots, 0.975\}$ on the indicated model. No per-model refitting is performed; predicted losses are computed from the constants fitted on the smaller-scale experiments in Sec. 4.

| | | $\delta^{\text{opt}}$ | | $\mathbb{E}[\delta]$ | |
|---|---|---|---|---|---|
| Granularity | Model | $R^2$ | MSE | $R^2$ | MSE |
| Layer-wise | Qwen-1.5-14B | 0.9837 | 0.1662 | 0.9770 | 0.3785 |
| | Qwen-3-14B | 0.9786 | 0.2726 | 0.9783 | 0.4021 |
| | LLaMA-13B | 0.9646 | 0.7787 | 0.9944 | 0.1683 |
| | Qwen-3-32B | 0.9778 | 0.2596 | 0.9771 | 0.3908 |
| Matmul-wise | Qwen-3-14B | 0.9940 | 0.2138 | 0.9970 | 3.1558 |
| | Qwen-1.5-32B | 0.9631 | 0.5597 | 0.9850 | 1.1031 |
| | Qwen-3-32B | 0.9935 | 0.2050 | 0.9969 | 2.7820 |

**Mixed-precision training** Mixed-precision quantization has also been adopted in training to reduce the large memory footprint of gradient descent, which requires the storage of optimizer states and gradients in addition to forward activations. It has been shown that the training process can tolerate aggressive quantization and correct quantization noise in some components. (Micikevicius et al., 2017) is the pioneering work proposing the storage of all weights, activations, and gradients in FP16, while updating a copy of weights in FP32. This work also proposed scaling up the forward pass loss and unscaling the gradient before the weight update to avoid under-utilization of the FP16 representable range, leading to half the memory requirement and speed-ups of 2–6$\times$ relative to FP32 training. Recently, more aggressive quantization has been studied for mixed-precision training. (Mellempudi et al., 2019) trains models with E5M2 FP8, maintaining a master copy of the weights in FP16, and dynamically adjusting the scaling factor every few iterations. Hybrid-FP8 (Sun et al., 2019) improves FP8 training by using E4M3 for forward propagation and E5M2 for backward propagation, leading to matching performance to models trained with FP32. Popular implementations of FP8 mixed-precision training like TransformerEngine[2] have achieved a training acceleration of around 3–4$\times$ compared to FP16 mixed-precision training.

**Hypotheses on other efficient-AI methods.** While we focused primarily on quantization, a natural direction for future work is to examine scaling trends for other efficient-AI methods such as sparsity. We hypothesize that the scaling laws for such methods closely resemble the laws for quantization introduced here. More broadly, we hypothesize the existence of *a broader scaling law governing how the ratio of approximate compute to exact compute scales with model size*, and the granularity at which approximate compute is applied.

[2]TransformerEngine: https://github.com/NVIDIA/TransformerEngine.