MIXED PRECISION SCALING LAWS FOR QUANTIZED PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Since scaling laws were introduced for modern LLMs (Kaplan et al., 2020), several works have extended them to account for quantization, tracking the emerging low-precision data types supported on AI accelerators. We develop a more accurate scaling law for quantized pretraining, grounded in optimization theory and validated empirically. Our formulation separates the effects of quantization on model capacity and on optimization efficiency. Through convergence analysis, we further observe that both forward and backward quantization can introduce an optimization error that persists with continued training, which we model explicitly in our law. To design our experimental regimes used to fit and evaluate our scaling law, we first revisit existing 4-bit training recipes and decompose them into their constituent forward and backward pass quantization choices, isolating the components that most strongly affect loss. Our Qwen3 experiments under selected 4-bit regimes, together with published Llama-style experiments, show that our law predicts loss with up to 11× lower MSE than existing laws as model size and token count scale. Finally, we extend our law to layer-wise mixed precision, and identify the optimal quantized-pretraining recipe, i.e., model size, training tokens, and per-layer forward/backward precision, under a hardware-aware compute budget.

1 INTRODUCTION

Scaling laws have played a central role in guiding large language model (LLM) pretraining by characterizing how training loss varies with model size, data, and compute (Kaplan et al., 2020; Hoffmann et al., 2022; Videau et al., 2026). By extrapolating from smaller scale experiments, they enable researchers to choose effective training configurations for expensive runs in advance. As pretraining costs continue to grow, accurate scaling laws are becoming increasingly valuable for allocating limited resources and maximizing model quality under a given compute budget.

Quantized training offers a promising path towards more efficient LLM pretraining by performing computations in the forward and backward passes at reduced numerical precision (Micikevicius et al., 2022; Fishman et al., 2025). Recent recipes push precision down to 4 bits using block-scaled formats such as NVFP4, while combining different techniques in the forward and backward passes to preserve training quality (Tseng et al., 2025; Abecassis et al., 2025). Existing methods differ both in how quantization scales are shared and selected during the forward pass and in how gradients are estimated during the backward pass. For example, Abecassis et al. (2025) and Panferov et al. (2026a) adopt different choices along both dimensions. Together, these techniques have substantially narrowed the gap between FP4 and BF16 training, making low-precision training an increasingly practical option for LLM pretraining (Chmiel et al., 2025; Abecassis et al., 2025).

We first revisit existing NVFP4 training recipes by decomposing them into forward and backward components and isolating the contribution of each to pretraining loss, since prior works typically proposed and evaluated these techniques only as complete configurations (Section 2). We find that forward quantization choices affect loss more than backward techniques, but even the best combination leaves a noticeable loss gap relative to BF16 training. Mixed-precision quantized training offers a promising way to narrow this gap, either by assigning different precisions to the forward and backward passes (Castro et al., 2026) or by varying precision across layers (Abecassis et al., 2025). Choosing among such configurations before training requires accurately predicting their loss

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

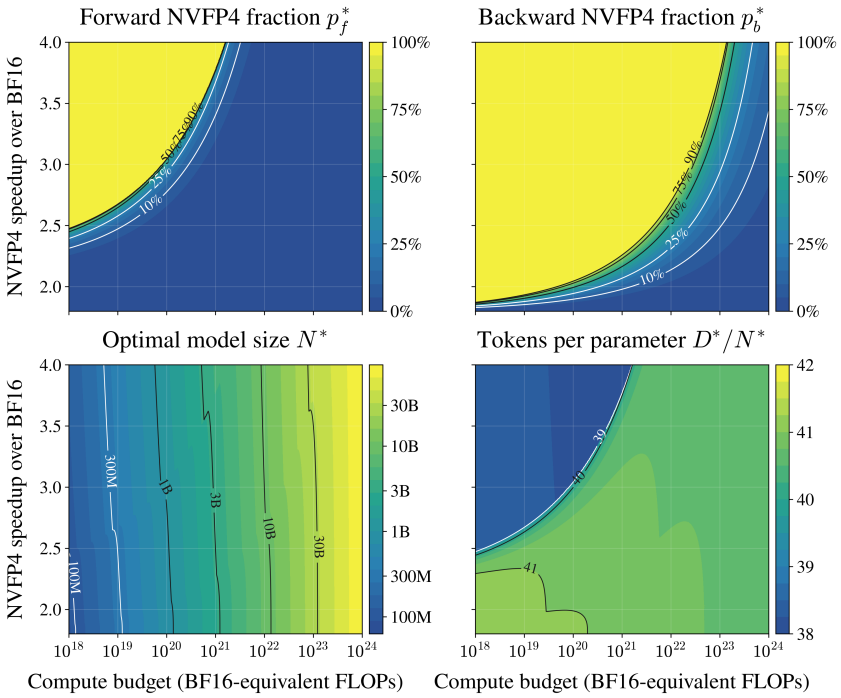

Figure 1: Compute-optimal precision allocation predicted by our scaling law. With each Transformer block using MXFP8 or NVFP4 in each pass, the panels show the optimal fraction of NVFP4 blocks in the forward pass p_f^* (top, left) and backward pass p_b^* (top, right), the optimal model size N^* (bottom, left), and tokens per parameter D^*/N^* (bottom, right), given the compute budget (x-axis) and the NVFP4 speedup over BF16 (y-axis), with the MXFP8 speedup fixed at $1.8\times$ (Section 4.3).

at the target model size and token budget, which motivates scaling laws for mixed precision both across the forward and backward passes and across layers.

For mixed precision across the forward and backward passes, existing scaling laws model forward quantization as a reduction in the effective parameter count (Kumar et al., 2025; Frantar et al., 2025; Panferov et al., 2026b), and the Quartet law additionally models backward quantization as a reduction in the effective training token count, whose penalty can be offset by training on more data (Castro et al., 2026) (Table 2). However, quantization also introduces optimization error that can persist with continued training. In the forward pass, evaluating gradients at quantized weights while updating full-precision weights can induce oscillations that prevent convergence (Bengio et al., 2013; Bai et al., 2019; Défossez et al., 2022; Nagel et al., 2022), and in the backward pass, quantization can bias gradient estimates (Ajalloeian & Stich, 2020). Accurate loss prediction therefore requires modeling this persistent error in both passes.

In this work, we develop a scaling law for quantized pretraining that accounts for this persistent optimization error (Section 3). Guided by a theoretical analysis of the optimization gap, the law combines effective parameter and data counts with separate additive terms for persistent optimization error from forward and backward quantization. The resulting law (Table 2) takes the form

$$L = \left(A\left(\rho_N(\psi_f)N\right)^{-\alpha} + B\left(\rho_D(\psi_b)D\right)^{-\beta}\right)^{\gamma} + \epsilon_f(\psi_f) + \epsilon_b(\psi_b) + E,$$

where ρ_N and ρ_D rescale the parameter count N and the training token count D, respectively, and ϵ_f and ϵ_b model persistent optimization error associated with the forward and backward quantization schemes ψ_f and ψ_b, respectively. We evaluate the law on our Qwen3 (Yang et al., 2025) pretraining experiments, covering models with up to 0.9B parameters trained on up to 35.8B tokens with BF16, MXFP8, and different NVFP4 schemes in both passes, and on the published Quartet results for Llama-style models trained with up to 800 tokens per parameter (Castro et al., 2026). Across both settings, the law predicts held-out loss more accurately than existing laws when extrapolating to larger model sizes and token budgets.

For mixed precision across layers, we extend the law to predict the loss when different quantization schemes are assigned to different Transformer blocks (Section 4). The extension interpolates between the predicted losses of the two schemes being mixed, with an exponent that captures the strategy used to select blocks for each scheme (Table 2). Accounting for the hardware speedup of each format, we then use the law to identify the compute-optimal model size, number of training tokens, and precision of the forward and backward passes under a given compute budget (Figure 1).

We summarize our main contributions as follows:

(a) We revisit existing NVFP4 recipes by decomposing them into forward and backward components, identifying which components matter for loss and measuring the remaining gap to BF16, which guides the design of our scaling law experiments (Section 2).

(b) We develop a scaling law for quantized pretraining with different forward and backward schemes that includes additive terms for persistent optimization error, backed by theoretical analysis (Table 2). It predicts held-out loss more accurately than existing laws on our Qwen3 experiments and the published Quartet results (Section 3).

(c) We extend the law to layer-wise mixed precision, and identify compute-optimal model sizes and precision assignments under a hardware-aware compute budget (Section 4).

2 Revisiting quantized pretraining recipes

Quantized pretraining reduces training cost by quantizing computation in both the forward and backward passes, with recent recipes pushing precision down to FP4 using block-scaled formats such as NVFP4. Since these recipes are typically evaluated as complete configurations, it remains unclear which of their individual forward and backward components matter for pretraining loss. In this section, we decompose two state-of-the-art FP4 recipes, the NVFP4 recipe (Abecassis et al., 2025) and Quartet II (Panferov et al., 2026a), into their forward and backward components to isolate which matter most for pretraining loss, and select the most effective regimes to adopt in our scaling law experiments. In the forward pass, the NVFP4 recipe uses *2D scaling*, sharing scales across 16×16 weight blocks so that quantized weights remain consistent when transposed in the backward pass, whereas Quartet II uses *1D scaling* with 1×16 blocks and *4/6 scaling*, which selects the lower-error of two candidate scales for each block (Cook et al., 2025). In the backward pass, the NVFP4 recipe applies *stochastic rounding (SR)* to the output activation gradients and *randomized Hadamard transforms (RHT)* to the operands of the weight gradient computation, similar to the MXFP4 recipe (Tseng et al., 2025), whereas Quartet II uses *MS-EDEN*, which combines randomized rotations with EDEN-based rescaling to obtain unbiased matrix multiplication.

Table 1: Validation loss of FP4 pretraining recipes decomposed into forward and backward components, for a 430M parameter Qwen3 model trained at 12.8B tokens. Loss increase is relative to the BF16 baseline. The last column indicates whether each combination is included in our scaling law grid.

Forward	Backward	Val. loss ↓	Increase ↓	In grid
BF16	BF16	3.245	–	✓
	MS-EDEN	3.257	+0.36%	✓
	RHT + SR	3.255	+0.32%	✓
	SR	3.268	+0.72%	✓
4/6 + 1D scaling	BF16	3.275	+0.94%	✓
	MS-EDEN	3.280	+1.07%	✓
	RHT + SR	3.282	+1.14%	✓
	SR	3.291	+1.42%	✓
2D scaling	BF16	3.284	+1.22%	✗
	RHT + SR	3.291	+1.41%	✗
	SR	3.300	+1.69%	✗

Results Table 1 reports the validation loss of a Qwen3 model (Yang et al., 2025) with 430M non-embedding parameters trained on 12.8B tokens for each combination of forward and backward components. The results show that forward pass quantization is generally more sensitive than backward pass quantization. From the BF16 baseline, quantizing only the backward pass to NVFP4 increases the loss by 0.32–0.72%, whereas quantizing only the forward pass increases it by 0.94–1.22%. Similarly, the advantage of Quartet II (4/6 + 1D scaling, MS-EDEN) over the NVFP4 recipe (2D scaling, RHT + SR) comes largely from the forward pass. Starting from Quartet II, replacing MS-EDEN with RHT + SR increases the loss by only 0.07 percentage points, whereas subsequently replacing 4/6 + 1D scaling with 2D scaling while keeping RHT + SR in the backward pass increases

Table 2: Comparison of scaling laws for quantization and compression. The last three columns indicate whether each law models different schemes in the forward and backward passes, different schemes across layers, and persistent optimization error. ρ_N and ρ_D rescale N and D according to the forward and backward schemes ψ_f and ψ_b, and ϵ_f and ϵ_b model persistent optimization error independent of N and D. For layer-wise mixed precision, $p \in [0, 1]$ is the fraction of layers assigned to the second of two schemes with losses L_1 and L_2, and γ_S depends on the layer selection strategy S. Other notation is described in Appendix E.

Scaling law	**Loss formulation**	**Fwd–bwd mixed**	**Layer mixed**	**Optim. error $\epsilon(\cdot)$**
Precision (PTQ) (Kumar et al., 2025)	$L = AN^{-\alpha} + BD^{-\beta} + E + C_T D^{\gamma_D} N^{-\gamma_N} e^{-P_{\text{post}}/\gamma_{\text{post}}}$	✗	✗	✗
Precision (training) (Kumar et al., 2025)	$L = A\left((1 - e^{-P/\bar{\gamma}})^3 N\right)^{-\alpha} + BD^{-\beta} + E$	✗	✗	✗
Compression (Frantar et al., 2025)	$L = A\left(\rho_N(\psi_f)N\right)^{-\alpha} + BD^{-\beta} + E$	✗	✗	✗
QAT (Chen et al., 2025)	$L = AN^{-\alpha} + BD^{-\beta} + E + kD^{\gamma_D} N^{-\gamma_N} (\log_2 G)^{\gamma_G}$	✗	✗	✗
Representations (Panferov et al., 2026b)	$L = A\left(N\,\widetilde{\rho}(\text{GMSE}(\psi_f))\right)^{-\alpha} + BD^{-\beta} + E$	✗	✗	✗
Quartet (Castro et al., 2026)	$L = \left(A\left(\rho_N(\psi_f)N\right)^{-\alpha} + B\left(\rho_D(\psi_b)D\right)^{-\beta}\right)^{\gamma} + E$	✓	✗	✗
Ours	(Section 3) $L = \left(A\left(\rho_N(\psi_f)N\right)^{-\alpha} + B\left(\rho_D(\psi_b)D\right)^{-\beta}\right)^{\gamma} + \epsilon_f(\psi_f) + \epsilon_b(\psi_b) + E,$ (Section 4) $L = (L_2 - L_1)\,p^{\gamma_S} + L_1$	✓	✓	✓

it by a further 0.27 percentage points. Based on these results, we adopt 4/6 + 1D scaling for NVFP4 forward quantization rather than 2D scaling, which performs worse overall, and retain all three backward techniques in our scaling law experiments. These selected regimes, together with BF16 and MXFP8 in both passes, form our experimental grid (Table 3).

Notably, even with the best FP4 recipe (Quartet II), a noticeable loss gap remains relative to the BF16 baseline, suggesting that uniformly quantizing to 4-bit may not be preferable. Mixed precision offers a way to narrow this gap by assigning different precisions to the forward and backward passes (Castro et al., 2026) or by varying precision across layers (Abecassis et al., 2025; Li & humans& ai, 2026). Accurately predicting the quantized pretraining loss of these configurations can help practitioners choose ones that meet their quality and compute targets, motivating quantized pretraining scaling laws for mixed precision across the forward and backward passes and across layers.

To this end, we develop a scaling law in Section 3, guided by theoretical analysis, that predicts quantized pretraining loss under different forward and backward quantization schemes more accurately than existing laws. In Section 4, we extend this law to layer-wise mixed precision and use it to identify compute-optimal configurations. We evaluate these laws on the selected regimes, and additionally validate the law in Section 3 on the published Quartet grid (Castro et al., 2026).

3 Quantized pretraining scaling laws with optimization error

Existing scaling laws for quantized pretraining treat forward quantization as reducing the effective parameter count (Kumar et al., 2025; Frantar et al., 2025; Panferov et al., 2026b) and backward quantization as reducing the effective training token count (Castro et al., 2026), implying that quantization penalties can be offset by scaling model size or data (Table 2). However, quantized training can also introduce optimization error that does not decrease with more data or parameters, as weights can oscillate across rounding boundaries in the forward pass (Défossez et al., 2022; Nagel et al., 2022) and gradient estimates can be biased in the backward pass (Ajalloeian & Stich, 2020).

We develop a scaling law that accounts for this persistent optimization error,

$$L = \left(A\left(\rho_N(\psi_f)N\right)^{-\alpha} + B\left(\rho_D(\psi_b)D\right)^{-\beta}\right)^{\gamma} + \epsilon_f(\psi_f) + \epsilon_b(\psi_b) + E, \tag{1}$$

where ρ_N and ρ_D rescale N and D according to the forward and backward schemes ψ_f and ψ_b, and $\epsilon_f, \epsilon_b \geq 0$ capture persistent optimization error introduced by each pass. Section 3.1 formalizes this intuition using optimization theory, and Sections 3.2 and 3.3 empirically validate that the resulting law predicts held-out loss more accurately than existing laws.

3.1 SCALING LAW FORMULATION

Quantization during pretraining affects both the representational capacity of the model and the accuracy of its training updates. To separate these effects, we decompose the loss gap into a capacity gap and an optimization gap (Su, 2026), which guides the formulation of our law. For a language model with parameters $\theta \in \mathbb{R}^N$, let $\mathcal{L}_N(\theta)$ denote the population loss. For a forward quantization scheme ψ_f, let $Q(\cdot;\psi_f) : \mathbb{R}^N \to \mathbb{R}^N$ denote the corresponding quantization map. We define the optimal loss of the quantized model and the irreducible loss over the model family, respectively, as

$$\mathcal{L}^*_{N,\psi_f} \coloneqq \inf_{\theta\in\mathbb{R}^N} \mathcal{L}_N(Q(\theta;\psi_f)), \qquad E \coloneqq \inf_N \inf_{\theta\in\mathbb{R}^N} \mathcal{L}_N(\theta) \qquad (\mathcal{L}^*_{N,\psi_f} \geq E).$$

Consider training for T steps, with weights θ_t and quantized weights $\hat{\theta}_t = Q(\theta_t;\psi_f)$ at step $t \in [T]$. The gap between the average expected loss of the quantized model and E decomposes as

$$\mathbb{E}_{J,\theta_J}\left[\mathcal{L}_N(\hat{\theta}_J)\right] - E = \underbrace{\left(\mathcal{L}^*_{N,\psi_f} - E\right)}_{\text{capacity gap}} + \underbrace{\left(\mathbb{E}_{J,\theta_J}\left[\mathcal{L}_N(\hat{\theta}_J)\right] - \mathcal{L}^*_{N,\psi_f}\right)}_{\text{optimization gap}},$$

where J is sampled uniformly from $[T]$. The first term represents the model *capacity gap* due to the limited number of parameters and the restricted numerical representations imposed by forward quantization. The second term represents the *optimization gap*, measuring how far the average expected loss during training remains above the best loss achievable by the quantized model.

Capacity gap. The capacity gap $\mathcal{L}^*_{N,\psi_f} - E$ depends only on model size N and the forward quantization ψ_f. The backward quantization ψ_b only affects training updates without changing the set of functions the model can represent. We therefore model it through the effective parameter count $\rho_N(\psi_f)N$, as in existing laws (Kumar et al., 2025; Castro et al., 2026; Panferov et al., 2026b).

Optimization gap. To examine how quantization affects optimization, Remark 3.1 below bounds the optimization gap for training with quantized weights and biased backward gradients under suitable regularity conditions. Similar analyses appear in Li et al. (2017); Panferov et al. (2026b) for quantized weights and Ajalloeian & Stich (2020) for biased gradients.

Remark 3.1. Consider updates $\theta_{t+1} = \theta_t - \eta\hat{g}_t$, where $\hat{g}_t \in \mathbb{R}^N$ is a minibatch gradient estimate at $\hat{\theta}_t = Q(\theta_t;\psi_f)$ using backward quantization scheme ψ_b. Define $b_t \coloneqq \mathbb{E}[\hat{g}_t \mid \theta_t] - \nabla\mathcal{L}_N(\hat{\theta}_t)$. For a minimizer q^* of $\mathcal{L}_N$ over $Q(\mathbb{R}^N;\psi_f)$, set $R_1(N,\psi_f) \coloneqq \mathbb{E}\|\theta_1 - q^*\|^2$. Assume ψ_f is deterministic, $\mathcal{L}_N$ is differentiable and convex, and, uniformly over t, T, $\mathbb{E}\|\nabla\mathcal{L}_N(\hat{\theta}_t)\|^2 \leq G^2$ and $\hat{g}_t$, b_t, $\hat{\theta}_t - \theta_t$, and $\theta_t - q^*$ have bounded second moments. For learning rate chosen as $\eta = c/\sqrt{T}$ $(c > 0)$,

$$\mathbb{E}_{J,\theta_J}[\mathcal{L}_N(\hat{\theta}_J)] - \mathcal{L}^*_{N,\psi_f} \leq \underbrace{\frac{1}{2\sqrt{T}}\left[\frac{R_1(N,\psi_f)}{c} + \frac{c}{T}\sum_{t=1}^{T}\mathbb{E}\|\hat{g}_t\|^2\right]}_{\text{(a) } O(T^{-1/2})}$$
$$+ \underbrace{G\left(\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\hat{\theta}_t - \theta_t\|^2\right)^{1/2}}_{\text{(b) } O(1)} + \underbrace{\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|b_t\|\|\theta_t - q^*\|]}_{\text{(c) } O(1)}.$$

Each term in the bound corresponds to a term in the scaling law (Eq. 1).

(a) The first term decreases as $O(T^{-1/2})$ with additional optimization steps, which correspond to training tokens at a fixed batch size. It motivates the effective data size $\rho_D(\psi_b)D$.

(b) The second term depends on the forward quantization error $\|\hat{\theta}_t - \theta_t\|$, which need not vanish with additional training. Example 3.1 shows that forward quantization can cause an $\Omega(1)$ optimization gap, motivating the persistent forward error term $\epsilon_f(\psi_f)$. Related oscillations and illustrative examples have been studied by Défossez et al. (2022); Nagel et al. (2022).

(c) The third term reflects bias b_t in backward gradient estimates, which can persist despite additional training (Ajalloeian & Stich, 2020), motivating the backward error term $\epsilon_b(\psi_b)$.

Table 3: Overview of the experimental data grids used to evaluate scaling laws for quantized pretraining. Model sizes N are non-embedding parameter counts. TPP denotes tokens per parameter (D/N), where D is the training token count. Token budgets show the range covered by each grid, with detailed settings in Appendix B.2. Each grid covers all combinations of the listed forward and backward schemes at every evaluated model size and token budget, including the BF16 baseline.

Data grid (architecture)	# points	Model sizes N	Token budgets	Forward schemes ψ_f	Backward schemes ψ_b
Quartet grid (Llama)	84	30M, 50M, 100M, 200M	25–800 TPP	BF16, MXFP4 (QuEST)	BF16, MXFP4 (SR)
Our grid (Qwen3)	375	280M, 430M, 640M, 910M	12.8B–35.8B	BF16, MXFP8, NVFP4 (4/6 + 1D)	BF16, MXFP8, NVFP4 (MS-EDEN), NVFP4 (RHT + SR), NVFP4 (SR)

Example 3.1 (Persistent optimization error from forward quantization). Consider $\mathcal{L}_1(\theta) = (\theta - \frac{1}{4})^2$ and $Q(\theta; \psi_f) = \lfloor \theta + \frac{1}{2} \rfloor$. For T divisible by four, initialize $\theta_1 = \frac{1}{2}$ and choose $\eta = c/\sqrt{T}$ with $0 < c \leq \frac{1}{4}$. Using exact gradients at the quantized weights $\hat{\theta}_t = Q(\theta_t; \psi_f)$ gives the updates $\theta_{t+1} = \theta_t - 2\eta(\hat{\theta}_t - \frac{1}{4})$. The quantized iterates repeat the cycle $(1, 0, 0, 0)$. The optimal quantized weight is $q^* = 0$, with $\mathcal{L}^*_{1,\psi_f} = \frac{1}{16}$. The optimization gap is

$$\mathbb{E}_J[\mathcal{L}_1(\hat{\theta}_J)] - \mathcal{L}^*_{1,\psi_f} = \frac{1}{4} \cdot \frac{9}{16} + \frac{3}{4} \cdot \frac{1}{16} - \frac{1}{16} = \frac{1}{8} = \Omega(1).$$

Thus, the optimization gap does not vanish as $T \to \infty$.

Proposed scaling law. Combining the capacity gap, modeled through $\rho_N(\psi_f)N$, with terms (a), (b), and (c) yields our proposed scaling law (Eq. 1), where the persistent contributions are modeled as separate additive terms. While term (a) depends on both quantization schemes, we retain only the backward dependence in ρ_D because including ψ_f did not improve held-out loss prediction (Section 3.3). Table 2 highlights the differences between our law and existing formulations.

3.2 Experimental setup

We evaluate our scaling law on two pretraining datasets: (1) the published Quartet experiments with Llama models (Castro et al., 2026) and (2) our Qwen3 experiments. For each grid, we fit the laws on calibration data and evaluate predictions on held-out runs with larger model sizes or token budgets.

Data grids. Table 3 details the two grids, the Quartet grid with 84 published runs of Llama-style models (Touvron et al., 2023) over two forward and two backward schemes (Castro et al., 2026; Panferov et al., 2025), and our grid with 375 runs of Qwen3-based models (Yang et al., 2025) over three forward and five backward schemes selected in Section 2 (see Appendix B.1 for details).

Evaluation protocol. We compare our law with the Quartet law (Castro et al., 2026) under *model size extrapolation*, fitting on the three smaller models and evaluating on the largest, and *data extrapolation*, fitting on smaller token budgets and evaluating at the largest (Appendix B.2). Following Castro et al. (2026), we fit each law in two stages, first to BF16 runs and then, with the baseline coefficients frozen, to quantized runs (Appendix B.3). We report R^2, mean squared error (MSE), and mean absolute percentage error (MAPE) on held-out quantized runs.

3.3 Experimental results

Table 4 reports model size and data extrapolation results on both grids. Our law outperforms the Quartet law across all settings, reducing MSE by 1.6× to 11×, e.g., from 1.6×10^{-4} to 1.4×10^{-5} (11×) for model size extrapolation on the Quartet grid and from 2.7×10^{-5} to 6.1×10^{-6} (4.4×) for data extrapolation on our grid. Figure 2 illustrates this for the held-out 200M Quartet models, where our predictions more closely track the measured losses across token budgets, showing that modeling persistent optimization error improves extrapolation to larger models and token budgets.

Ablation results. We further ablate the design choices in Eq. 1 by evaluating three variants, (1) allowing ρ_D to depend on both forward and backward quantization schemes, (2) removing the backward error term $\epsilon_b(\psi_b)$, and (3) removing the forward error term $\epsilon_f(\psi_f)$. Table 5 reports the geometric mean of held-out MSE across model size and data extrapolation on the Quartet grid and our grid,

Table 4: Held-out loss prediction under model size and data extrapolation on the Quartet grid (Llama) and our grid (Qwen3). Each law is fitted to the calibration split of each setting and evaluated on the quantized runs in the held-out split. The best results are shown in bold.

		Quartet grid (Llama)			Our grid (Qwen3)		
Evaluation	Scaling law	$R^2 \uparrow$	MSE $\downarrow$	MAPE (%) $\downarrow$	$R^2 \uparrow$	MSE $\downarrow$	MAPE (%) $\downarrow$
Model size extrapolation	Quartet	0.964	1.6e-4	0.386	0.990	8.4e-6	0.076
	Ours	**0.997**	**1.4e-5**	**0.096**	**0.994**	**5.3e-6**	**0.059**
Data extrapolation	Quartet	0.995	9.0e-5	0.234	0.992	2.7e-5	0.121
	Ours	**0.997**	**5.1e-5**	**0.217**	**0.998**	**6.1e-6**	**0.064**

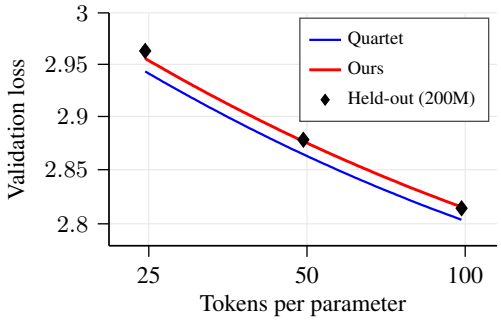

Figure 2: Model size extrapolation to held-out 200M models on the Quartet grid with MXFP4 QuEST forward quantization and stochastic rounding in the backward pass.

Table 5: Ablations of our scaling law, reporting geometric-mean held-out MSE across model size and data extrapolation on the Quartet and Qwen3 grids. Variants modify our law as indicated. Best result is bolded.

Scaling law form	Geo. mean MSE $\downarrow$
Quartet	4.27e-5
Ours	**1.23e-5**
- (1) With $\rho_D(\psi_f, \psi_b)$	1.29e-5
- (2) Without $\epsilon_b(\psi_b)$	1.34e-5
- (3) Without $\epsilon_f(\psi_f)$	3.76e-5

giving equal weight to relative changes across settings with different error scales. The proposed law achieves the lowest error among the evaluated variants, supporting the chosen formulation.

4 Layer-wise mixed-precision pretraining

Practical quantized pretraining recipes often retain higher precision in sensitive layers while using lower-precision computation elsewhere, enabling a flexible trade-off between training loss and cost (Abecassis et al., 2025; Li & humans& ai, 2026; Pan et al., 2026). Selecting such configurations before training requires predicting how loss depends not only on the fraction of high-precision blocks, but also on which blocks are assigned higher precision. In this section, we extend Eq. 1 to model both the fraction of high-precision blocks and how they are selected (Table 2), and use the extended law to identify compute-optimal precision allocations. Section 4.1 formulates the extension, Section 4.2 evaluates its predictions, and Section 4.3 studies compute-optimal precision allocation under a fixed compute budget, showing the optimal precision increases with the budget.

4.1 Formulation

Consider a model with K Transformer blocks and two quantization configurations: high precision ψ_1 and low precision ψ_2, each comprising forward and backward schemes, $\psi_i = (\psi_{f,i}, \psi_{b,i})$ for $i \in \{1, 2\}$. We interpolate between the two configurations by assigning ψ_2 to a subset of the blocks and ψ_1 to the rest. A selection strategy $S : [0, 1] \to 2^{[K]}$ maps a fraction p to a set of blocks $S(p) \subseteq [K]$ with $|S(p)| = \text{round}(pK)$, and the mixed configuration $\psi_S^{\text{mix}}(p)$ assigns ψ_2 to the blocks in $S(p)$ and ψ_1 to the remaining blocks. We model its loss as

$$L\big(N, D, \psi_S^{\text{mix}}(p)\big) = (L_2 - L_1)\, p^{\gamma_S} + L_1, \tag{2}$$

where $L_i = L(N, D, \psi_{f,i}, \psi_{b,i})$ is predicted by Eq. 1, $L_2 \geq L_1$, and $\gamma_S > 0$ depends on the strategy S. Mixing only the forward or only the backward corresponds to $\psi_{b,1} = \psi_{b,2}$ or $\psi_{f,1} = \psi_{f,2}$.

By construction, the loss equals L_1 at $p = 0$ and L_2 at $p = 1$, so the strategy only determines how the loss moves between these endpoints. A good strategy assigns ψ_2 to the least sensitive blocks first, keeping the loss close to L_1 until p becomes large, whereas a poor strategy assigns it to sensitive

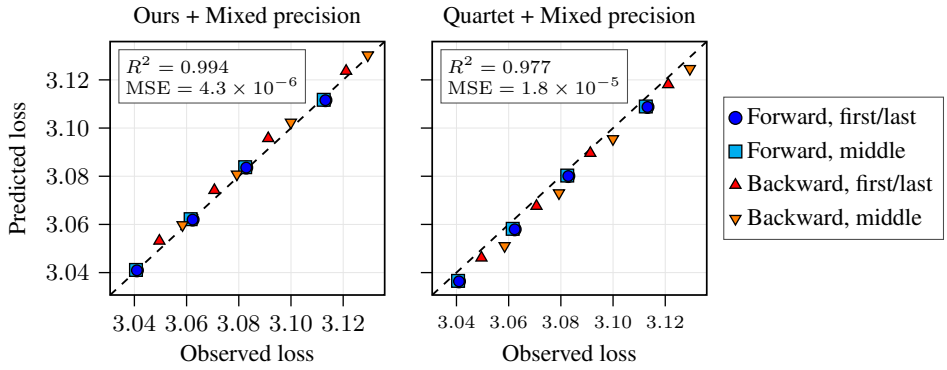

Figure 3: Predicted versus observed loss of layer-wise mixed-precision runs on the held-out 910M model, with γ_S fitted on the three smaller models. The endpoints L_1 and L_2 in Eq. 2 are predicted by our law (left) or the Quartet law (right). Markers indicate the mixed pass and the blocks kept in higher precision, and each point corresponds to one token budget.

blocks early and increases the loss quickly. The exponent γ_S captures this difference, where a larger γ_S indicates that the strategy keeps the sensitive blocks in ψ_1 longer, a smaller γ_S indicates that it assigns them to ψ_2 earlier, and $\gamma_S = 1$ corresponds to linear interpolation between L_1 and L_2.

4.2 Experiments

We evaluate the extended law on Qwen3 models with layer-wise MXFP8/NVFP4 mixed precision. We vary precision in either the forward or backward pass, assigning half the blocks to NVFP4 ($p = 0.5$). We compare two strategies for selecting higher-precision blocks: (1) *first/last*, which keeps the first and last blocks in MXFP8, following practical recipes (Abecassis et al., 2025; Li & humans& ai, 2026), and (2) *middle*, which keeps the central blocks in MXFP8. We hold out the 910M model, predict endpoints L_1 and L_2 using laws fitted on smaller models, and fit γ_S for each pass and strategy using their mixed-precision runs (see Appendix B.4 for details).

Prediction accuracy. Figure 3 compares the predicted and observed losses of the mixed-precision runs on the held-out 910M model. With endpoints predicted by our law, the extended law closely matches the observed losses across both passes, both strategies, and all token budgets ($R^2 = 0.994$, MSE $= 4.3 \times 10^{-6}$), indicating that γ_S fitted on smaller models transfers to a larger model. With endpoints predicted by the Quartet law, the predictions consistently underestimate the observed losses, increasing the MSE by about four times ($R^2 = 0.977$, MSE $= 1.8 \times 10^{-5}$).

First and last blocks matter more in the backward pass. Table 6 reports the fitted γ_S, where larger values indicate that a strategy retains more sensitive blocks at higher precision and incurs a smaller fraction of the loss gap. Keeping the first and last blocks in higher precision, as proposed in practical NVFP4 training recipes (Abecassis et al., 2025; Li & humans& ai, 2026), is particularly effective in the backward pass, as reflected in its higher fitted γ_S. In the forward pass, both strategies incur about 45% of the gap at $p = 0.5$, so placement has little effect. In the backward pass, keeping the first and last blocks in MXFP8 incurs 37.6% of the gap, compared with 69.5% for the middle blocks.

Table 6: Fitted γ_S (higher means better strategy) with our law, and the resulting fraction of the loss gap incurred at $p = 0.5$.

Pass	Strategy S	$\gamma_S \uparrow$	Gap at $p = 0.5 \downarrow$
Forward	First/last	1.16	44.7%
	Middle	1.14	45.5%
Backward	First/last	1.41	37.6%
	Middle	0.53	69.5%

4.3 Compute-optimal precision allocation

Using the scaling laws from Sections 3 and 4.1, we identify the compute-optimal forward, backward and layer-wise precision, along with model size and the number of training tokens, under a given compute budget (Figure 1). Unlike prior precision scaling laws (Kumar et al., 2025, Section 4.3.2), our law predicts that the optimal precision increases with the compute budget.

Setup. We consider the setting of mixing (1) MXFP8 and (2) NVFP4 with 4/6 + 1D scaling in the forward pass, and (1) MXFP8 and (2) NVFP4 with MS-EDEN in the backward pass, assigning NVFP4 to fractions $p_f \in [0,1]$ and $p_b \in [0,1]$ of the blocks, respectively, with the ***first/last*** strategy. Applying Eq. 2 to each pass in turn, we predict the loss as

$$L = (1-u)(1-v)\,L_{\mathrm{MX8,MX8}} + u(1-v)\,L_{\mathrm{NV4,MX8}} + (1-u)v\,L_{\mathrm{MX8,NV4}} + uv\,L_{\mathrm{NV4,NV4}}, \quad (3)$$

where $u = p_f^{\gamma_f}$, $v = p_b^{\gamma_b}$, $L_{f,b}$ is the loss predicted by Eq. 1 with forward format f and backward format b, and γ_f and γ_b are the first/last exponents in Table 6. We assume that each exponent is independent of the NVFP4 backward technique and of the format in the other pass, which makes Eq. 3 independent of the order in which the passes are mixed.

Cost model and optimal allocation. The forward and backward passes account for roughly one third and two thirds of the $6ND$ training FLOPs, respectively (Kaplan et al., 2020). Let $\lambda_{f,\mathrm{MX8}}$, $\lambda_{f,\mathrm{NV4}}$, $\lambda_{b,\mathrm{MX8}}$, and $\lambda_{b,\mathrm{NV4}}$ denote the hardware-dependent speedups of MXFP8 and NVFP4 quantization recipes over BF16 in each pass. Assuming equal cost across blocks and ignoring operations other than matrix multiplications, the training cost in BF16-equivalent FLOPs is

$$C(N, D, p_f, p_b) = 2ND\left(\frac{1-p_f}{\lambda_{f,\mathrm{MX8}}} + \frac{p_f}{\lambda_{f,\mathrm{NV4}}}\right) + 4ND\left(\frac{1-p_b}{\lambda_{b,\mathrm{MX8}}} + \frac{p_b}{\lambda_{b,\mathrm{NV4}}}\right). \quad (4)$$

Given a compute budget C_0, the *compute-optimal allocation* solves

$$\min_{N,\,D,\,p_f,\,p_b} L(N, D, p_f, p_b) \quad \text{subject to} \quad C(N, D, p_f, p_b) \le C_0, \quad p_f, p_b \in [0,1], \quad (5)$$

where L is given by Eq. 3. We solve this problem via numerical optimization (Appendix D).

Figure 1 shows the solution of Eq. 5, specifically the optimal model size N^*, the optimal number of training tokens D^*, and the optimal fractions of blocks in NVFP4 in the forward and backward passes, p_f^* and p_b^*, as a function of the compute budget C_0 (x-axis) and the NVFP4 speedup λ_{NV4} (y-axis). We assume that each format has the same speedup in the forward and backward passes, i.e., $\lambda_{f,\mathrm{MX8}} = \lambda_{b,\mathrm{MX8}} = \lambda_{\mathrm{MX8}}$ and $\lambda_{f,\mathrm{NV4}} = \lambda_{b,\mathrm{NV4}} = \lambda_{\mathrm{NV4}}$, and fix $\lambda_{\mathrm{MX8}} = 1.8$, following the GEMM kernel speedup reported by NVIDIA Transformer Engine (NVIDIA, 2026).

Optimal precision increases with the budget. Following a horizontal line in the top panels of Figure 1, i.e., fixing the NVFP4 speedup λ_{NV4} while increasing the compute budget C_0, the optimal NVFP4 fractions p_f^* (top left) and p_b^* (top right) decrease, leaving more blocks in MXFP8. For example, at $\lambda_{\mathrm{NV4}} = 3$, the optimal uniform configuration switches to MXFP8 in the forward pass around $C_0 = 10^{20}$ FLOPs and in both passes at around $C_0 = 6 \times 10^{22}$ FLOPs, while layer-wise mixing smooths these switches into gradual transitions. This contrasts with prior scaling laws, which predict budget-independent optimal precision when jointly optimizing model size, data, and precision (Kumar et al., 2025, Section 4.3.2), and stems from the persistent optimization error in our law. As C_0 grows, model- and data-dependent loss terms shrink while persistent error remains, so the lower error of higher precision eventually outweighs the compute savings of lower precision.

The backward pass should use low precision first. Comparing the top panels of Figure 1 at a fixed NVFP4 speedup, the backward pass keeps NVFP4 up to much larger compute budgets than the forward pass. Three factors favor NVFP4 in the backward pass. The backward pass accounts for two thirds of the training FLOPs, so each block in NVFP4 saves more compute. Its larger exponent γ_b (Table 6) means that assigning blocks to NVFP4 incurs a smaller fraction of the loss gap. Finally, the persistent error of NVFP4 relative to MXFP8 is about half as large in the backward pass as in the forward pass in our fitted law (Appendix C). Practitioners with a limited budget for higher precision should therefore allocate it to the forward pass first.

5 Conclusion

In this work, we develop a scaling law guided by theoretical analysis to predict the loss of quantized pretraining with different forward and backward quantization schemes, modeling persistent optimization error from each pass, and show that it predicts held-out loss more accurately than existing laws. Extending the law to layer-wise mixed precision, we find that keeping the first and last blocks in higher precision is particularly effective in the backward pass. Finally, the law predicts that the compute-optimal precision increases with the compute budget, unlike previous precision scaling laws, since the persistent optimization error does not diminish with scale.

AI USE STATEMENT

We used large language models and coding agents to assist with drafting and editing the manuscript, implementing plotting and optimization code, and providing assistance with mathematical derivations and proofs. We reviewed and verified the AI-assisted content and take full responsibility for the paper.

ETHICS STATEMENT

This work studies scaling laws and optimization for quantized language model pretraining. It does not involve sensitive personal data or real-world deployment. While improved training efficiency may broaden access to capable models, it may also enable harmful downstream uses. We encourage responsible use of these methods.

REPRODUCIBILITY STATEMENT

We provide the experimental settings, model configurations, quantization schemes, optimization details, and fitting procedures needed to reproduce our results. We also report the scaling-law experiments and ablations used in our analyses.

REFERENCES

Felix Abecassis, Anjulie Agrusa, Dong Ahn, Jonah Alben, Stefania Alborghetti, Michael Andersch, Sivakumar Arayandi, Alexis Bjorlin, Aaron Blakeman, Evan Briones, et al. Pretraining large language models with nvfp4. *arXiv preprint arXiv:2509.25149*, 2025.

Ahmad Ajalloeian and Sebastian U Stich. On the convergence of sgd with biased gradients. *arXiv preprint arXiv:2008.00051*, 2020.

Yu Bai, Yu-Xiang Wang, and Edo Liberty. Proxquant: Quantized neural networks via proximal operators. *ICLR*, 2019.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. Smollm-corpus, 2024. URL `https://huggingface.co/datasets/HuggingFaceTB/smollm-corpus`.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Johan Bjorck, Alon Benhaim, Vishrav Chaudhary, Furu Wei, and Xia Song. Scaling optimal lr across token horizons. In *International Conference on Learning Representations*, volume 2025, pp. 83640–83657, 2025.

Dan Busbridge, Amitis Shidani, Floris Weers, Jason Ramapuram, Etai Littwin, and Russell Webb. Distillation scaling laws. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 5977–6045. PMLR, 13–19 Jul 2025.

Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208, 1995.

Roberto Castro, Andrei Panferov, Rush Tabesh, Oliver Sieberling, Jiale Chen, Mahdi Nikdan, Saleh Ashkboos, and Dan Alistarh. Quartet: Native fp4 training can be optimal for large language models. *Advances in Neural Information Processing Systems*, 38:43552–43572, 2026.

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

Brian Chmiel, Maxim Fishman, Ron Banner, and Daniel Soudry. Fp4 all the way: Fully quantized training of llms. *arXiv preprint arXiv:2505.19115*, 2025.

Jack Cook, Junxian Guo, Guangxuan Xiao, Yujun Lin, Keith Wyss, Mahdi Nazemi, Asit Mishra, Carlo del Mundo, Tijmen Blankevoort, and Song Han. Four over six: More accurate nvfp4 quantization with adaptive block scaling. *arXiv preprint arXiv:2512.02010*, 2025.

Alexandre Défossez, Yossi Adi, and Gabriel Synnaeve. Differentiable model compression via pseudo quantization noise. *TMLR*, 2022.

Nolan Dey, Bin Zhang, Lorenzo Noci, Mufan Li, Blake Bordelon, Shane Bergsma, Cengiz Pehlevan, Boris Hanin, and Joel Hestness. Don't be lazy: Completep enables compute-efficient deep transformers. *Advances in Neural Information Processing Systems*, 38:137707–137739, 2026.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 1286–1305, 2021.

Maxim Fishman, Brian Chmiel, Ron Banner, and Daniel Soudry. Scaling fp8 training to trillion-token llms. In *International Conference on Learning Representations*, volume 2025, pp. 98631–98644, 2025.

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

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. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

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

Tanishq Kumar, Zachary Ankner, Benjamin Spector, Blake Bordelon, Niklas Muennighoff, Mansheej Paul, Cengiz Pehlevan, Christopher Ré, and Aditi Raghunathan. Scaling laws for precision. In *International Conference on Learning Representations*, volume 2025, pp. 71833–71864, 2025.

Hao Li, Soham De, Zheng Xu, Christoph Studer, Hanan Samet, and Tom Goldstein. Training quantized nets: A deeper understanding. *Advances in neural information processing systems*, 30, 2017.

Ziang Li and humans& ai. The 4-bitter lesson: Balancing stability and performance in nvfp4 rl. *humans& ai Blog*, July 2026. https://humansand.ai/blog/nvfp4-rl.html.

Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5(5):5, 2017.

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

Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. Overcoming oscillations in quantization-aware training. In *International Conference on Machine Learning*, pp. 16318–16330. PMLR, 2022.

NVIDIA. Transformer engine, 2026. URL `https://github.com/NVIDIA/TransformerEngine/blob/9c33764bd9fd88cb4984ef2b60e9f0623055d3b2/docs/examples/gemm_profiling/gemm_profiling.rst`.

Yunjie Pan, Yongyi Yang, Hanmei Yang, and Scott Mahlke. Snip: An adaptive mixed precision framework for subbyte large language model training. In *Proceedings of the 31st ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pp. 1815–1831, 2026.

Andrei Panferov, Jiale Chen, Soroush Tabesh, Mahdi Nikdan, and Dan Alistarh. QuEST: Stable training of LLMs with 1-bit weights and activations. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 47820–47836. PMLR, 13–19 Jul 2025.

Andrei Panferov, Erik Schultheis, Soroush Tabesh, and Dan Alistarh. Quartet ii: Accurate llm pre-training in nvfp4 by improved unbiased gradient estimation. *arXiv preprint arXiv:2601.22813*, 2026a.

Andrei Panferov, Alexandra Volkova, Ionut-Vlad Modoranu, Vage Egiazarian, Mher Safaryan, and Dan Alistarh. Unified scaling laws for compressed representations. *Advances in Neural Information Processing Systems*, 38:151086–151112, 2026b.

Jianlin Su. Deconstructing scaling laws: The interplay of optimization, architecture, and data, July 2026. URL `https://kexue.fm/archives/11833`.

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

Albert Tseng, Tao Yu, and Youngsuk Park. Training llms with mxfp4. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan (eds.), *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 1630–1638. PMLR, 03–05 May 2025.

Mathurin Videau, Badr Youbi-Idrissi, David Lopez-Paz, and Kartik Ahuja. Skaling: Chinchilla's exponents meet kaplan's coupling. *arXiv preprint arXiv:2608.07222*, 2026.

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

A Proof of Remark 3.1

In this section, we provide the proof of Remark 3.1, restated here for convenience.

Remark 3.1. Consider updates $\theta_{t+1} = \theta_t - \eta \hat{g}_t$, where $\hat{g}_t \in \mathbb{R}^N$ is a minibatch gradient estimate at $\hat{\theta}_t = Q(\theta_t; \psi_f)$ using backward quantization scheme ψ_b. Define $b_t \coloneqq \mathbb{E}[\hat{g}_t \mid \theta_t] - \nabla \mathcal{L}_N(\hat{\theta}_t)$. For a minimizer q^* of $\mathcal{L}_N$ over $Q(\mathbb{R}^N; \psi_f)$, set $R_1(N, \psi_f) \coloneqq \mathbb{E}\|\theta_1 - q^*\|^2$. Assume ψ_f is deterministic, $\mathcal{L}_N$ is differentiable and convex, and, uniformly over t, T, $\mathbb{E}\|\nabla \mathcal{L}_N(\hat{\theta}_t)\|^2 \le G^2$ and $\hat{g}_t$, b_t, $\hat{\theta}_t - \theta_t$, and $\theta_t - q^*$ have bounded second moments. For learning rate chosen as $\eta = c/\sqrt{T}$ $(c > 0)$,

$$\mathbb{E}_{J,\theta_J}[\mathcal{L}_N(\hat{\theta}_J)] - \mathcal{L}^*_{N,\psi_f} \le \underbrace{\frac{1}{2\sqrt{T}} \left[\frac{R_1(N,\psi_f)}{c} + \frac{c}{T} \sum_{t=1}^{T} \mathbb{E}\|\hat{g}_t\|^2 \right]}_{\text{(a) } O(T^{-1/2})}$$
$$+ \underbrace{G \left(\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\|\hat{\theta}_t - \theta_t\|^2 \right)^{1/2}}_{\text{(b) } O(1)} + \underbrace{\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[\|b_t\| \|\theta_t - q^*\|]}_{\text{(c) } O(1)}.$$

Proof. Write $g_t = \nabla \mathcal{L}_N(\hat{\theta}_t)$ and $e_t = \hat{\theta}_t - \theta_t$. By convexity and $\mathbb{E}[\hat{g}_t \mid \theta_t] = g_t + b_t$,

$$\begin{aligned} \mathbb{E}[\mathcal{L}_N(\hat{\theta}_t) - \mathcal{L}_N(q^*)] &\le \mathbb{E}\langle g_t, \theta_t - q^* + e_t \rangle \\ &= \mathbb{E}\langle \hat{g}_t, \theta_t - q^* \rangle - \mathbb{E}\langle b_t, \theta_t - q^* \rangle + \mathbb{E}\langle g_t, e_t \rangle. \end{aligned}$$

The update rule gives

$$2\eta \langle \hat{g}_t, \theta_t - q^* \rangle = \|\theta_t - q^*\|^2 - \|\theta_{t+1} - q^*\|^2 + \eta^2 \|\hat{g}_t\|^2.$$

Summing over t and dropping the nonpositive terminal term yields

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[\mathcal{L}_N(\hat{\theta}_t) - \mathcal{L}_N(q^*)] \le \frac{R_1(N,\psi_f)}{2\eta T} + \frac{\eta}{2T} \sum_{t=1}^{T} \mathbb{E}\|\hat{g}_t\|^2 \\ + \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[\|b_t\| \|\theta_t - q^*\|] + \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\langle g_t, e_t \rangle. \end{aligned}$$

By Cauchy–Schwarz,

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\langle g_t, e_t \rangle \le \left(\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\|g_t\|^2 \right)^{1/2} \left(\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\|e_t\|^2 \right)^{1/2} \le G \left(\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\|e_t\|^2 \right)^{1/2}.$$

Taking J uniformly from $\{1, \ldots, T\}$ independently of the trajectory, and substituting $\eta = c/\sqrt{T}$ and $\mathcal{L}_N(q^*) = \mathcal{L}^*_{N,\psi_f}$, proves the claim. The stated orders follow from the uniform second-moment bounds. □

B Implementation details

B.1 Model architecture and training

Our scaling law grid models are derived from the Qwen3-8B architecture (Yang et al., 2025) by varying the number of Transformer blocks K and setting the remaining dimensions from K following the proportions of Qwen3-8B, which has $K = 36$ blocks, width $d_{\text{model}} = 4096$, and 32 query heads. The width scales linearly with depth, $d_{\text{model}} = \text{round}(4096K/36,\ 128)$, and the number of query heads scales with width, $n_{\text{heads}} = \text{round}(32\, d_{\text{model}}/4096,\ 8)$, where $\text{round}(x, m)$ rounds x to the nearest multiple of m. All other choices follow Qwen3-8B, including grouped-query attention with 8 key-value heads and a head dimension of 128, an FFN dimension of $3d_{\text{model}}$, QK normalization, and tied input and output embeddings over the Qwen3 vocabulary of 151,936 tokens. Table 7 lists the resulting configurations, where non-embedding parameter counts include normalization weights.

Table 7: Model configurations derived from Qwen3-8B. K denotes the number of Transformer blocks. Parameter counts are given without and with the tied embeddings.

Model	K	d_{model}	Query heads	KV heads	FFN dim.	Non-emb. params	Total params
280M	12	1408	8	8	4224	283M	497M
430M	14	1536	16	8	4608	429M	663M
640M	16	1792	16	8	5376	639M	911M
910M	18	2048	16	8	6144	906M	1217M
Qwen3-8B	36	4096	32	8	12288	6946M	7568M

Table 8: Calibration and held-out splits for model size and data extrapolation. # points denotes the number of data points in each split, including all forward and backward scheme combinations. The Quartet grid uses tokens-per-parameter ratios (D/N), whereas our grid uses training token counts (D). N denotes the non-embedding parameter count.

		Calibration		Held-out	
Data grid	Evaluation	# points	Model sizes and token budgets	# points	Model sizes and token budgets
Quartet grid (Llama)	Model size extrapolation	72	$N \in \{30, 50, 100\}$M $D/N \in \{25, 50, 100, 200, 400, 800\}$	12	$N = 200$M $D/N \in \{25, 50, 100\}$
	Data extrapolation	60	$N \in \{30, 50, 100\}$M $D/N \in \{25, 50, 100, 200, 400\}$	12	$N \in \{30, 50, 100\}$M $D/N = 800$
Our grid (Qwen3)	Model size extrapolation	315	$N \in \{280, 430, 640\}$M $D \in \{12.8, 15.3, 17.9, 21.1, 25.5, 30.0, 35.8\}$B	60	$N = 910$M $D \in \{18.1, 21.7, 25.4, 29.9\}$B
	Data extrapolation	270	$N \in \{280, 430, 640\}$M $D \in \{12.8, 15.3, 17.9, 21.1, 25.5, 30.0\}$B	45	$N \in \{280, 430, 640\}$M $D = 35.8$B

Parameterization and initialization We use the CompleteP parameterization (Dey et al., 2026) with Qwen3-8B as the base shape, so that hyperparameters tuned on one model transfer across width and depth. For each model size, all runs start from the same initial checkpoint and see the same data order, so BF16 and all quantized recipes differ only in their quantization schemes.

Optimization and data We use AdamW (Loshchilov et al., 2017) ($\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0.1, gradient clipping 1.0) with batches of 1024 sequences of length 4096 (about 4.2M tokens), and a warmup–stable–decay schedule (Hu et al., 2024) with 244 warmup steps and a linear decay to 1% of the peak over the final 10% of steps. A BF16 sweep on the 280M model at 5.7B tokens (20 tokens per parameter) gives an optimal peak learning rate of 1.2×10^{-3}, which is also optimal for the 170M and 430M models, confirming transfer across width and depth (Dey et al., 2026). Scaling it to the smallest token budget of each model (Table 8) with $\eta \propto D^{-0.32}$ (Bjorck et al., 2025) gives 9.25×10^{-4} for the three smaller models and 8.27×10^{-4} for the 910M model, which we use for all budgets and quantization schemes. We train on SmolLM-Corpus (Ben Allal et al., 2024), with the Qwen3 tokenizer, and report validation loss on C4 (Dodge et al., 2021).

Token budgets We train every model with at least 20 tokens per parameter (Hoffmann et al., 2022), using the token budgets listed in Table 8. The three smaller models share the budgets of the 640M model, ranging from 12.8B to 35.8B tokens, and the 910M model uses budgets from 18.1B to 29.9B tokens. For cost efficiency, we obtain each token budget by resuming from the run of a smaller budget before its decay phase and applying its own 10% decay.

B.2 Evaluation splits

Table 8 lists the calibration and held-out splits for the two extrapolation settings in Section 3.2. In model size extrapolation, each law is fitted to runs with the three smaller models of each grid and evaluated on the largest model, which is 200M for the Quartet grid and 910M for our grid. In data extrapolation, each law is fitted to runs with the three smaller models at all but the largest token budget and evaluated at that budget, which is 800 tokens per parameter for the Quartet grid and 35.8B tokens for our grid. The largest model is therefore excluded from data extrapolation. In both settings, the laws are fitted only to the calibration split.

B.3 Fitting the scaling laws

We fit both our law and the Quartet law in two stages, following Castro et al. (2026), and both stages minimize a Huber loss with threshold 10^{-3} on log-loss residuals $\log \hat{L} - \log L$. We use Levenberg–Marquardt with 200 initializations for the baseline fit and a bounded trust-region reflective solver with 20 initializations for the quantization parameters. We repeat both stages independently for each data grid and extrapolation setting using only calibration data.

B.4 Layer-wise mixed-precision experiments

We use four Qwen3 model sizes with 280M, 430M, 640M, and 910M non-embedding parameters, each trained with four token budgets, yielding 16 combinations of N and D. The three smaller models are trained on 12.8B, 15.3B, 17.9B, and 21.1B tokens, and the largest model on 18.1B, 21.7B, 25.4B, and 29.9B tokens. We mix precision in one pass at a time.

- *Forward mixing.* ψ_1 and ψ_2 use MXFP8 and NVFP4 with 4/6 + 1D scaling in the forward pass, respectively, and both use MS-EDEN in the backward pass.
- *Backward mixing.* ψ_1 and ψ_2 use MXFP8 with round-to-nearest and NVFP4 with SR in the backward pass, respectively, and both use 4/6 + 1D scaling in the forward pass.

We evaluate the extended law under model size extrapolation, holding out the 910M model. To predict the endpoints L_1 and L_2, we use each scaling law fitted to the calibration split for model size extrapolation on our grid (Appendix B.2), which excludes the 910M model. We then fit γ_S for each pass and strategy by minimizing the MSE between Eq. 2 and the observed losses of the mixed-precision runs on the three smaller models, so that the 910M model is excluded from both fits.

C Fitted scaling law parameters

Table 9 lists the parameters of our scaling law (Eq. 1) fitted to the calibration split of each grid and extrapolation setting. The shared coefficients E, A, B, α, β, and γ are fitted in the first stage to the BF16 runs, and the scheme-dependent parameters ρ_N, ρ_D, ϵ_f, and ϵ_b are fitted in the second stage to the quantized runs, with $\rho_N = \rho_D = 1$ and $\epsilon_f = \epsilon_b = 0$ for BF16 by construction. N and D are measured in non-embedding parameters and tokens, respectively.

D Solving the compute-optimal allocation

The mixed loss in Eq. 3 combines the four uniform endpoint losses $L_{f,b}(N, D)$ from Eq. 1, all evaluated at the same N and D, since a mixed configuration trains a single model on a single dataset. We write the cost in Eq. 4 as $C = ND\, c(p_f, p_b)$, where

$$c(p_f, p_b) = 2\left(\frac{1-p_f}{\lambda_{f,\mathrm{MX8}}} + \frac{p_f}{\lambda_{f,\mathrm{NV4}}}\right) + 4\left(\frac{1-p_b}{\lambda_{b,\mathrm{MX8}}} + \frac{p_b}{\lambda_{b,\mathrm{NV4}}}\right) \tag{6}$$

is the training cost per parameter per token, which equals 6 for BF16. Since the loss decreases in D, the budget constraint is tight at the optimum, so $D = C_0/(N\, c(p_f, p_b))$, and we solve

$$\min_{p_f,\, p_b \in [0,1],\ N>0} L\left(N,\ \frac{C_0}{N\, c(p_f, p_b)},\ p_f,\ p_b\right). \tag{7}$$

This problem does not admit a closed-form solution, since the endpoint losses have different optimal splits between parameters and data. We therefore solve it numerically, similar to Kumar et al. (2025, Appendix E.3), who solve a related compute-optimal problem via grid search. Specifically, we perform a grid search over (p_f, p_b, N) and refine the best candidates with L-BFGS-B (Byrd et al., 1995), running it from multiple starting points and keeping the best solution, as in the fitting procedure of Hoffmann et al. (2022, Section 3.3).

Table 9: Fitted parameters of our scaling law on the Quartet grid (Llama) and our grid (Qwen3), for each extrapolation setting (model size and data). The shared coefficients are fitted to the BF16 runs, and the scheme-dependent parameters to the quantized runs, with $\rho_N = \rho_D = 1$ and $\epsilon_f = \epsilon_b = 0$ for BF16 by construction. N and D are measured in non-embedding parameters and tokens.

		Quartet grid (Llama)		Our grid (Qwen3)	
Parameter	Scheme	Model size	Data	Model size	Data
E	–	1.650	1.763	2.546	2.574
A	–	1.10×10^5	3.50×10^5	7.62×10^4	9.38×10^4
B	–	2.50×10^5	6.70×10^5	8.21×10^5	1.04×10^6
α	–	0.622	0.701	0.629	0.642
β	–	0.550	0.605	0.631	0.642
γ	–	0.322	0.315	0.767	0.787
ρ_N	MXFP4 (QuEST)	0.770	0.798	–	–
	MXFP8	–	–	0.987	0.984
	NVFP4 (4/6 + 1D)	–	–	0.922	0.923
ρ_D	MXFP4 (SR)	0.951	0.973	–	–
	MXFP8	–	–	0.998	1.000
	NVFP4 (MS-EDEN)	–	–	1.000	1.000
	NVFP4 (RHT + SR)	–	–	1.000	1.000
	NVFP4 (SR)	–	–	1.000	1.000
ϵ_f	MXFP4 (QuEST)	0.0340	0.0397	–	–
	MXFP8	–	–	0.0021	0.0016
	NVFP4 (4/6 + 1D)	–	–	0.0151	0.0152
ϵ_b	MXFP4 (SR)	0.0076	0.0100	–	–
	MXFP8	–	–	0.0004	0.0008
	NVFP4 (MS-EDEN)	–	–	0.0076	0.0076
	NVFP4 (RHT + SR)	–	–	0.0076	0.0076
	NVFP4 (SR)	–	–	0.0215	0.0211

E Related work

Kaplan et al. (2020) propose a coupled power law of the form $L = (AN^{-\alpha} + BD^{-1})^{\gamma}$, where the outer exponent γ couples the model and data terms and the inner data exponent is fixed to one. The Chinchilla scaling law (Hoffmann et al., 2022) takes the additive form $L = AN^{-\alpha} + BD^{-\beta} + E$, allowing separate exponents for model size and data but no interaction between their contributions. More recent work combines independent inner exponents with an outer coupling exponent in the form $L = (AN^{-\alpha} + BD^{-\beta})^{\gamma} + E$, empirically improving loss prediction across data regimes (Busbridge et al., 2025; Videau et al., 2026).

Table 2 summarizes existing laws that incorporate quantization or compression. For post-training quantization (PTQ), Kumar et al. (2025) add a degradation term $C_T D^{\gamma_D} N^{-\gamma_N} e^{-P_{\text{post}}/\gamma_{\text{post}}}$ to the Chinchilla law, where P_{post} denotes the post-training precision. The degradation grows with the training token count and decreases with model size and precision. For quantized training, they replace N with the effective parameter count $(1-e^{-P/\bar{\gamma}})^3 N$, where P denotes the training precision and $\bar{\gamma}$ is a fitted constant. Frantar et al. (2025) generalize this rescaling to an effective parameter count $\rho_N(\psi_f)N$ that depends on the compression scheme. For quantization-aware training (QAT), Chen et al. (2025) add an error term $kD^{\gamma_D} N^{-\gamma_N} (\log_2 G)^{\gamma_G}$ that also depends on the quantization group size G. Panferov et al. (2026b) predict parameter efficiency from the representation error of the compression scheme, where $\mathrm{GMSE}(\psi_f)$ measures the error of fitting Gaussian data using the forward scheme ψ_f and $\widetilde{\rho}$ maps this error to parameter efficiency. Finally, the Quartet law (Castro et al., 2026) rescales both the parameter count and the training token count, with $\rho_N(\psi_f)$ depending on the forward scheme and $\rho_D(\psi_b)$ on the backward scheme.