BEYOND SHADOW WEIGHTS: QUANTIZATION-AWARE TRAINING AS QUANTIZED-ENDPOINT DESCENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Quantization-aware training (QAT) updates a full-precision shadow weight $\mathbf{x}$ but deploys the quantized endpoint $Q(\mathbf{x})$. Existing explanations for QAT largely view its success through the lens of shadow weights: QAT can move $\mathbf{x}$ toward flatter basins, gain robustness from oscillations, or balance the shadow loss $f(\mathbf{x})$ against the quantization error $\|\mathbf{x} - Q(\mathbf{x})\|_2$. These perspectives do not directly explain the empirical observation that the deployed endpoint loss $f(Q(\mathbf{x}))$ improves even as the shadow loss $f(\mathbf{x})$ increases substantially. In this paper, we offer a different explanation by treating QAT as finite-grid endpoint dynamics. Motivated by the approximate normality of rescaled pretrained weights, we propose an idealized residual-phase model that yields a crossing law that determines which coordinates cross quantization boundaries after a shadow update. We also find that signal imbalance creates asymmetric oscillations that favor weaker-gradient endpoints. This motivates *QAR (Quantization with Amplified Routing)*, an algorithmic framework that directly operates on the quantization code. We further prove feasible-gradient guarantees for QAR with a family of power amplifier. Post-training experiments on three large language models support the endpoint view and show that QAR can achieve performance comparable to QAT with up to **24.1**% peak memory saving.

1 INTRODUCTION

Modern deep neural networks have achieved broad empirical success across language, vision, and decision-making tasks (Brown et al., 2020; He et al., 2016; Silver et al., 2016), by scaling parameters, data, and compute (Kaplan et al., 2020; Hoffmann et al., 2022). But this scaling comes at a cost. As the model becomes larger, deployment is increasingly constrained by memory footprint, and inference slows down due to high memory bandwidth and compute throughput. This phenomenon is highly pronounced for large language models (LLMs). Low-bit quantization addresses this issue by replacing high-precision weights with a small set of representable values. Post-training quantization methods compress a pretrained model with little additional training, and they have become a central tool for LLM deployment. Representative examples include GPTQ (Frantar et al., 2022), SmoothQuant (Xiao et al., 2023), and AWQ (Lin et al., 2024). However, at more aggressive bit widths, the quantized model can lose accuracy substantially. In this regime, quantization-aware training (QAT) becomes the standard method for repair. For example, LLM-QAT (Liu et al., 2024) and EfficientQAT (Chen et al., 2025) make this repair step practical for LLMs with data-free distillation or reduced memory cost.

Conceptually, QAT tries to solve the deployment problem. Let $f : \mathbb{R}^d \to \mathbb{R}$ be the full-precision loss, Q_b be a b-bit quantizer, and $\mathcal{Q}_b$ be the finite set of deployed endpoints. To achieve the best deployment performance, the training objective should be a discrete constrained problem

$$\min_{\mathbf{q} \in \mathcal{Q}_b} f(\mathbf{q}). \tag{1}$$

By introducing a full-precision shadow weight $\mathbf{x}$, we can equivalently write the above problem as

$$\min_{\mathbf{x} \in \mathbb{R}^d} F_b(\mathbf{x}) = f(Q_b(\mathbf{x})).$$

Since Q_b is constant on each quantization cell, the exact gradient of $f(Q_b(\mathbf{x}))$ is zero almost everywhere, direct differentiation therefore gives no useful descent direction. QAT bypasses this

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.

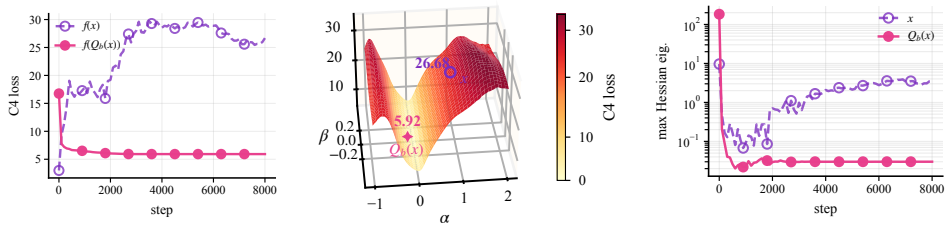
degeneracy through the straight-through estimator (STE), which treats Q_b as the identity in the backward pass (Bengio et al., 2013). In the notation above, one step of QAT has the form

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \mathbf{u}_t, \qquad \mathbf{u}_t = \mathcal{U}_t(\nabla f(Q_b(\mathbf{x}_t))),$$

where $\mathcal{U}_t(\cdot)$ is a time-dependent update map. STE-based QAT has been effective in practice (Courbariaux et al., 2015; Zhou et al., 2016; Jacob et al., 2018; Choi et al., 2018; Esser et al., 2020), but it remains largely heuristic for solving the problem (1). From a classical optimization viewpoint, QAT is unusual because the variable being updated is not the variable being deployed. During training, QAT maintains a full-precision shadow weight $\mathbf{x}$, also called the latent weight, but at deployment the model uses the quantized endpoint $\mathbf{q}$. Thus, to analyze QAT, **the central analytical question is not how the shadow trajectory behaves, but how the deployed endpoint trajectory evolves.**

The main analytical difficulty lies in the discreteness induced by quantization. Standard first-order arguments alone fail to match the motion of the deployed endpoint. By L-smoothness of f, the classical gradient update $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t)$ yields $f(\mathbf{x}_{t+1}) - f(\mathbf{x_t}) \leq -\eta\big(1 - \frac{L\eta}{2}\big)\|\nabla f(\mathbf{x_t})\|_2^2$, so choosing η small enough guarantees a descent in the objective value. However, the loss at the QAT endpoint $f(Q_b(\mathbf{x}_t))$ behaves differently. Although the change in shadow weights $\mathbf{x}_{t+1} - \mathbf{x}_t$ is proportional to the stepsize η, once $\mathbf{x}_{t+1}$ crosses a quantization boundary, the change in the deployed endpoint $Q_b(\mathbf{x}_{t+1}) - Q_b(\mathbf{x}_t)$ is a grid jump. Shrinking η cannot reduce the size of this jump, hence descent of the endpoint loss no longer holds.

Existing explanations. For QAT with weight-only quantization, explanations in the existing literature can be grouped into three categories. The first is a flatness view: QAT can drive the shadow weight into a flat region where replacing $\mathbf{x}$ by $Q_b(\mathbf{x})$ changes the loss only mildly. For example, Javed et al. (2025) model quantization as adding noise for the shadow weight. In this way, QAT has an implicit regularizer for the second-order term and can steer the shadow point toward flatter regions. This perspective is inspired by the broader literature which links flatter regions of the loss landscape to better generalization (Hochreiter & Schmidhuber, 1997; Keskar et al., 2017; Jiang et al., 2020; Foret et al., 2021). A related explanation builds on the river-valley picture of loss landscapes (Wen et al., 2025). For instance, Li et al. (2026) argue that when the deployed endpoint leaves a low-loss basin, the STE update can include an inward component that pulls the quantized endpoint back toward the basin, where $f(Q_b(\mathbf{x}))$ stays close to $f(\mathbf{x})$. The second view interprets QAT as minimizing the shadow loss $f(\mathbf{x})$ plus a regularizer that penalizes distance of the shadow weight to the quantized set. ProxQuant (Bai et al., 2019) proposes a continuous regularized problem that encourages the shadow weight to approach the quantized set, and STE-based QAT corresponds to the hard-projection limiting case of the proximal framework. The third view emphasizes oscillation robustness gained from quantization. In a toy model, Wenshøj et al. (2025) argue that the STE update contains a residual-driven component that pushes shadow weights away from their nearest quantization level and toward cell boundaries. And this effect can be achieved by adding a regularization term to the shadow loss $f(\mathbf{x})$.



(a) 2-bit evaluation loss **(b)** Loss landscape at the final step **(c)** 2-bit max Hessian eigenvalue

Figure 1: Qwen-2.5-0.5B 2-bit loss and curvature diagnostics. (a) contrasts the increasing shadow loss with the low endpoint loss. (b) plots evaluation loss on the plane $Q_b(\mathbf{x}) + \alpha(\mathbf{x} - Q_b(\mathbf{x})) + \beta\|\mathbf{x} - Q_b(\mathbf{x})\|_2\mathbf{v}$ for a random unit $\mathbf{v} \perp (\mathbf{x} - Q_b(\mathbf{x}))$. (c) shows curvature better aligned with the deployed endpoint in 2 bits.

Figure 1 illustrates a regime that is not described by any account requiring the shadow loss $f(\mathbf{x}_t)$ to decrease. In this 2-bit Qwen-2.5-0.5B run, the loss at the shadow weight $f(\mathbf{x}_t)$ increases substantially while the deployed endpoint loss $f(Q_b(\mathbf{x}_t))$ remains much lower. Moreover, the loss landscape around the deployed endpoint $Q_b(\mathbf{x}_t)$ appears to be flatter, rather than around $\mathbf{x}_t$. These observations

suggest a stronger conclusion: QAT can improve the deployed quantized endpoint, even at the expense of sacrificing the performance of the full-precision objective. This motivates us to analyze the deployed endpoint loss directly.

Contribution. Our contributions are two-fold. First, we analyze QAT through the residual $\mathbf{r}_t = \mathbf{x}_t - Q_b(\mathbf{x}_t)$, which records the position of the shadow point inside its quantization cell. Motivated by the approximate normality of rescaled pretrained weights and the near-uniform residual phases observed at higher bit widths, we introduce an idealized conditional-uniform phase model. Under this model, endpoint updates admit a representation in terms of random boundary-crossing indicators. This representation yields an expected endpoint-descent bound that separates the first-order gain from aligned crossings from the second-order penalty induced by finite grid jumps, thereby identifying a regime in which QAT improves the deployed endpoint.

Second, our analysis further identifies signal imbalance as a source of asymmetric oscillations that favor weaker-gradient endpoints. Based on this mechanism, we develop **QAR** (**Q**uantization with **A**mplified **R**outing), a memory-efficient endpoint-training framework that updates quantized codes directly and prioritizes transitions associated with stronger signals. We establish convergence guarantee for QAR and characterize when signal amplification remains beneficial under stochastic gradient noise. Post-training experiments on Qwen-2.5-0.5B, Llama-3.2-3B, and Llama-2-7B support finite-grid endpoint view and show that QAR can achieve performance comparable to QAT with up to **24.1%** lower peak training memory.

2 PRELIMINARIES

To begin with, quantization maps high-precision floating-point values to low-precision discrete representations, thereby reducing memory footprint. We use the standard uniform affine quantization notation (Nagel et al., 2021). Throughout the paper, unless stated otherwise, we use a signed symmetric uniform b-bit quantizer $Q_b(\cdot)$.[1] Let $\mathbf{s} = (s_1, s_2, \ldots, s_d) > 0$ be the fixed local scale. Define $K_b = 2^{b-1} - 1$ and $\mathcal{K}_b = \{-K_b, -K_b+1, \ldots, K_b\}$. For shadow weight $\mathbf{x}$, the i-th coordinate of the deployed endpoint is

$$Q_b(\mathbf{x})_i = s_i \ \mathrm{clamp}\left(\left\lfloor \frac{x_i}{s_i} \right\rceil ; -K_b, K_b \right),$$

where $\lfloor \cdot \rceil$ denotes rounding to the nearest integer and $\mathrm{clamp}(a; \ell, u) = \min\{\max\{a, \ell\}, u\}$. When the bit width is clear from context, we write $Q(\mathbf{x})$ for $Q_b(\mathbf{x})$ for simplicity. Then the deployed values in coordinate i are $s_i\mathcal{K}_b$, with adjacent interior spacing s_i. For an interior code $k \in \{-K_b + 1, \ldots, K_b - 1\}$, the non-saturated cell is $C_{i,k} = \left[s_i\left(k - \frac{1}{2}\right), s_i\left(k + \frac{1}{2}\right) \right)$. We call coordinate i non-saturated when its deployed code $k_i = Q(\mathbf{x})_i / s_i$ satisfies $-K_b < k_i < K_b$; coordinates with $k_i = \pm K_b$ are saturated.

Let $\mathbf{g} = \nabla f(\mathbf{q})$ be the endpoint gradient. A one-step QAT shadow update with stepsize η is

$$\mathbf{x}^+ = \mathbf{x} - \eta \mathbf{u} \quad \text{with } \mathbf{u} = \mathcal{U}_t(\mathbf{g}). \tag{2}$$

Here $\mathcal{U}_t(\cdot)$ is the update rule applied to the endpoint gradient at step t. For example, GD takes $\mathcal{U}_t(\mathbf{g}) = \mathbf{g}$ and signGD corresponds to $\mathcal{U}_t(\mathbf{g}) = \mathrm{sign}(\mathbf{g})$. We further write $\mathbf{q}^+ = Q(\mathbf{x}^+)$, and call coordinate i active in this step when $u_i \neq 0$. Throughout, we use the convention $\mathrm{sign}(0) = 0$.

Assumption 1 (Bounded update direction)**.** *There exists a constant $U < \infty$ such that the update direction $\mathbf{u} = \mathcal{U}_t(\mathbf{g})$ satisfying $\|\mathbf{u}\|_\infty \leq U$.*

Assumption 2 (Smoothness)**.** *The loss f is differentiable, and there exists a nonnegative vector $\mathbf{L} = (L_1, \ldots, L_d)$ such that for all $\mathbf{x}, \mathbf{y}$,*

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{1}{2} \sum_{i=1}^{d} L_i (y_i - x_i)^2.$$

This coordinatewise smoothness is commonly used in sign-gradient analyses such as Bernstein et al. (2018). Standard L-smoothness implies this assumption by taking $L_i = L$ for all coordinates.

[1] In this convention, we drop the most negative code so that the positive and negative endpoints have equal magnitude. The resulting codebook has $2^b - 1$ levels. For example, the $b = 2$ setting in this paper has levels $\{-1, 0, 1\}$, so it corresponds to the ternary setting, or 1.58-bit, in common terminology.

Lemma 1 (First-order descent alignment)**.** *For the one-step QAT update (2) at point* $\mathbf{x}$ *with stepsize* η*, we have* $\mathbf{u} \odot (\mathbf{q}^+ - \mathbf{q}) \le 0$*. If in addition* $\mathbf{g} \odot \mathbf{u} \ge 0$*, then* $\mathbf{g} \odot (\mathbf{q}^+ - \mathbf{q}) \le 0$*, and consequently,*

$$\langle \mathbf{g}, \mathbf{q}^+ - \mathbf{q} \rangle \le 0.$$

Here $\odot$ *denotes the Hadamard product, and the inequalities should be understood coordinatewise.*

The proof is deferred to Appendix D.1. Lemma 1 shows that each realized endpoint jump moves opposite to the update direction. If the update is coordinatewise aligned with the endpoint gradient, the jump is also aligned with first-order descent. However, to get the descent of the loss, we need a more fine-grained bound. To achieve the bound, we assume that the stepsize is small enough that $\eta|u_i| < s_i$ for every coordinate i. In this case, each coordinate crosses at most one interior threshold per step, so $(\mathbf{q}^+ - \mathbf{q})_i \in \{-s_i, 0, s_i\}$. Let $\mathbf{J} = (J_1, \dots, J_d)$ with $J_i = \mathbf{1}\{q_i^+ \ne q_i\}$ be the crossing indicator, where $J_i = 1$ indicates that a one-cell endpoint jump occurs. Now we can describe the grid-jump mechanism of the QAT.

Proposition 2 (Realized endpoint loss bound)**.** *Suppose Assumptions 1 and 2 hold, and the stepsize is small enough that* $\eta U < \min_i s_i$*. Then*

$$\mathbf{q}^+ - \mathbf{q} = -\mathbf{s} \odot \operatorname{sign}(\mathbf{u}) \odot \mathbf{J},$$

and

$$f(\mathbf{q}^+) - f(\mathbf{q}) \le -\sum_{\{i:J_i=1\}} s_i \operatorname{sign}(u_i) g_i + \frac{1}{2} \sum_{\{i:J_i=1\}} L_i s_i^2. \tag{3}$$

Proposition 2 illustrates the finite-grid descent mechanism: once the crossed coordinates are known, the bound certifies endpoint descent whenever the first-order signal exceeds the second-order penalty. The proof is deferred to Appendix D.2.

3 Residual Homogeneity and Endpoint Motion

The bound in Proposition 2 controls the endpoint-loss change for a realized crossing set, but it does not specify how often each coordinate crosses. To quantitatively understand how the crossing mechanism of QAT aggregates to a steady decrease of the endpoint loss and to illustrate when QAT stops making progress, we also need to know how often such crossings occur. Thus, we model the crossing indicators J_i conditionally on the current endpoint and update direction. The resulting conditional laws provide an approximate perspective on QAT by linking the frequency of boundary crossings to expected descent at the quantized endpoint.

Model 1 (Conditional crossing model)**.** *Fix an endpoint* $\mathbf{q} = Q(\mathbf{x})$*, an update direction* $\mathbf{u}$*, and a stepsize* η *in a regime where each active coordinate can cross at most one neighboring quantization boundary. Let* $J_i \in \{0, 1\}$ *be the crossing indicator defined above. We model* J_i*, conditional on* $(\eta, \mathbf{q}, \mathbf{u})$*, as a Bernoulli random variable with conditional mean*

$$p_i(\eta, \mathbf{q}, \mathbf{u}) = \mathbb{E}[J_i \mid \eta, \mathbf{q}, \mathbf{u}] = \Pr(J_i = 1 \mid \eta, \mathbf{q}, \mathbf{u}).$$

The deterministic realized case is included by taking $p_i(\eta, \mathbf{q}, \mathbf{u}) \in \{0, 1\}$*.*

To further illustrate the inner structure of the endpoint motion, we introduce the residual phase. For $\mathbf{q} = Q(\mathbf{x})$, define the residual $\mathbf{r} = \mathbf{x} - Q(\mathbf{x})$. In a non-saturated cell, $|r_i| \le \frac{s_i}{2}$, and the normalized residual $\theta_i = r_i/s_i$ lies in $\left[-\frac{1}{2}, \frac{1}{2}\right)$. This phase is the state variable that decides whether the next shadow update changes the deployed endpoint. If x_i is near the center of its cell, a small shadow update leaves q_i unchanged; if x_i is within a boundary window of width $\eta|u_i|$, the same update produces an endpoint jump. We make this precise in Lemma 6 (see Appendix D.3): the crossing event is exactly the event that θ_i falls in a window of normalized width $\rho_i = \eta|u_i|/s_i$ at the boundary the update moves toward, so p_i is the mass the conditional phase law assigns to that window.

In LLM quantization, it is empirically observed that after layerwise or blockwise rescaling, the central mass of pretrained weights is often close to zero-centered normal distributions. QLoRA (Dettmers et al., 2023) uses this observation to motivate NormalFloat (NF4), a low-bit datatype designed for normally distributed weights. This observation is also related to mean-field analyses of wide neural-network training, where the empirical measure of the parameter admits a large-width

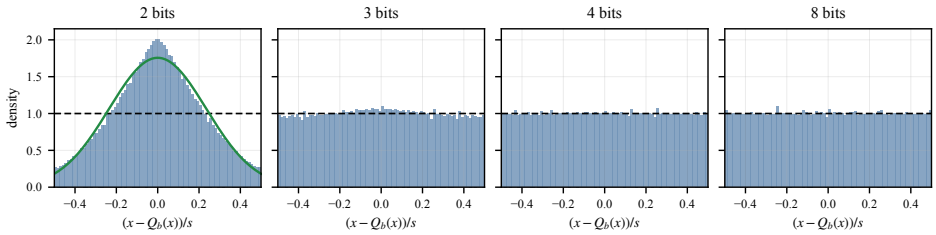


Figure 2: Distributions of the normalized residual $\theta_i = (x_i - Q(\mathbf{x})_i)/s_i$ for Qwen-2.5-0.5B at the pretrained point, pooled over non-saturated linear weights. The solid curve in the 2-bit panel is a fitted zero-centered truncated normal. The dashed reference is the uniform cell-phase density on $[-\frac{1}{2}, \frac{1}{2})$. The 3-, 4-, and 8-bit cases are close to the uniform residual-phase calculation underlying Model 2.

limiting description and cross-particle dependence becomes weak under suitable scaling (Sirignano & Spiliopoulos, 2020; De Bortoli et al., 2020).

Based on this observation, we reason about the residual phase induced by folding weights through a finite grid. Consider a scalar pretrained weight $X \sim \mathcal{N}(0, \tau^2)$ with $\tau > 0$, and a symmetric b-bit quantizer with grid spacing $s_b > 0$. When the weights are measured in units of one quantization step, X/s_b has standard deviation $\lambda_b = \tau/s_b$. Aggregating the normalized residual phase across all non-saturated interior cells gives a folded density

$$\pi(\theta) \propto \sum_{k=-K_b+1}^{K_b-1} \exp\left(-\frac{(k+\theta)^2}{2\lambda_b^2}\right), \qquad \theta \in \left[-\frac{1}{2}, \frac{1}{2}\right).$$

For this zero-centered folded model, the mass near the upper and lower cell boundaries is the same. We therefore write the one-sided boundary mass as $B(\rho) = \int_{1/2-\rho}^{1/2} \pi(\theta)\,\mathrm{d}\theta = \int_{-1/2}^{-1/2+\rho} \pi(\theta)\,\mathrm{d}\theta$ for $0 \le \rho < 1$. For $b = 2$, we have $K_2 = 1$, so the only non-saturated interior cell is the zero cell and the pooled residual phase is center-concentrated. Such a phase places less mass near the cell boundary than the uniform case, so $B(\rho) < \rho$ for small ρ. For $b > 2$, pooling over many interior cells makes the folded sum nearly flat within one cell, so $B(\rho) \approx \rho$. This is why the 3-, 4-, and 8-bit histograms in Figure 2 are much closer to a uniform cell-phase density. The arguments above start from a normally distributed weight parameter, but the resulting near-uniformity holds for a broad class of weight densities (see Appendix C.1 for more discussion). In Appendix C.2, we also report the predicted folded densities together with their empirical counterparts.

To obtain a tractable description throughout training, we introduce an approximation in which the residual phase remains representative after conditioning on the current endpoint: among non-saturated coordinates, the phase laws do not vary systematically with $\mathbf{q}$. This approximation implies that the update-selected crossing probabilities exhibit the same near-uniform behavior. Figure 3 gives the corresponding smooth-density intuition: conditioning on $\mathbf{q}$ restricts each active coordinate x_i to its own quantization cell centered at q_i. If the conditional marginal density of x_i varies little across that cell, then the phase $\theta_i = (x_i - q_i)/s_i$ is approximately uniform on $[-\frac{1}{2}, \frac{1}{2})$. Since the update-selected boundary window has normalized width $\rho_i = \eta|u_i|/s_i$, the uniform phase calculation gives crossing probability ρ_i. This gives the simplified conditional model.

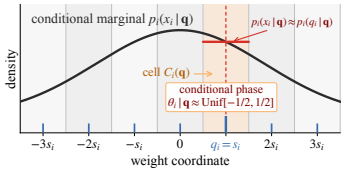


Figure 3: Illustration for residual homogeneity.

Model 2 (Simplified crossing model)**.** *Suppose Model 1 holds and that, conditional on the current endpoint* $\mathbf{q} = Q(\mathbf{x})$*, update* $\mathbf{u}$*, and stepsize* η*, each non-saturated coordinate has crossing probability*

$$p_i(\eta, \mathbf{q}, \mathbf{u}) = \Pr(J_i = 1 \mid \eta, \mathbf{q}, \mathbf{u}) = \frac{\eta|u_i|}{s_i}.$$

With the crossing model, we can translate random boundary crossings into an expected one-step bound on the endpoint loss, and the proof is deferred to Appendix D.4.

Theorem 3 (Conditional endpoint descent under the crossing model)**.** *Suppose Assumptions 1 and 2 hold, the crossings follow Model 1, and the stepsize* η *satisfies* $\eta U < s_i$ *for every coordinate* i*. Given the endpoint* $\mathbf{q} = Q(\mathbf{x})$ *and update direction* $\mathbf{u} = \mathcal{U}(\nabla f(\mathbf{q}))$*, write* $p_i = p_i(\eta, \mathbf{q}, \mathbf{u})$*. The endpoint*

update can be written as

$$\mathbf{q}^+ - \mathbf{q} = -\mathbf{s} \odot \operatorname{sign}(\mathbf{u}) \odot \mathbf{J} \tag{4}$$

where $J_i \in \{0, 1\}$ *satisfies* $\Pr(J_i = 1 \mid \eta, \mathbf{q}, \mathbf{u}) = p_i$. *Consequently,*

$$\mathbb{E}[f(\mathbf{q}^+) - f(\mathbf{q}) \mid \eta, \mathbf{q}, \mathbf{u}] \le -\sum_i p_i s_i \operatorname{sign}(u_i) g_i + \frac{1}{2} \sum_i p_i L_i s_i^2. \tag{5}$$

If, in addition, Model 2 holds and $\mathbf{q}$ *is non-saturated in each coordinate, then*

$$\mathbb{E}[f(\mathbf{q}^+) - f(\mathbf{q}) \mid \eta, \mathbf{q}, \mathbf{u}] \le -\eta \langle \mathbf{g}, \mathbf{u} \rangle + \frac{\eta}{2} \sum_i L_i s_i |u_i|. \tag{6}$$

Theorem 3 is the one-step certificate for quantized-endpoint descent. Under the uniform crossing model, the first-order term $-\eta \langle \mathbf{g}, \mathbf{u} \rangle$ matches that of continuous optimization. However, the second-order term is also linear in η, rather than $O(\eta^2)$ as in continuous optimization. This difference arises because a smaller η reduces the probability of a grid jump but not the size of an accepted jump. Therefore, making the stepsize arbitrarily small does not by itself guarantee descent. The bound therefore certifies descent when the first-order signal is large enough to dominate the second-order effect of one-cell jumps.

4 QAR: A Direct Endpoint Algorithmic Framework

4.1 Signal-Imbalance Induced Oscillation in QAT Dynamics

So far, we have used the crossing model to describe how shadow updates induce transitions between deployed endpoints. The simplified crossing model is reasonably well calibrated at higher bit widths (Figure 12 in Appendix B.3). However, this pooled view can overlook the temporal dependence in QAT dynamics. In particular, adjacent endpoints can have different update signals, causing the shadow weight to spend unequal amounts of time at them. Consider $f(q) = \frac{1}{2}(q - q^\star)^2$, adjacent endpoints $q_1 < q^\star < q_2$, and the shadow update $x_{t+1} = x_t - \eta g(Q(x_t))$, where $g(q) = q - q^\star$. If π_j denotes the residence fraction at q_j, the small-step two-cell dynamics approximately balance as

$$\pi_1 g(q_1) + \pi_2 g(q_2) \approx 0, \qquad \frac{\pi_2}{\pi_1} \approx \frac{|g(q_1)|}{|g(q_2)|}.$$

Thus, the weaker the signal at an endpoint, the longer the shadow resides there. In Figure 4, the shadow drifts slowly away from q_1, but returns rapidly after crossing to q_2. Such signal-imbalanced oscillations have been observed in QAT (Nagel et al., 2022).

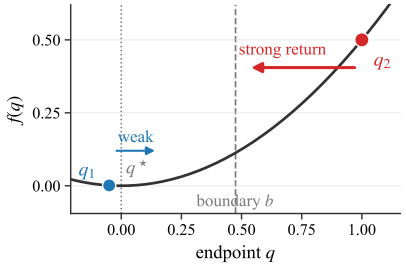


Figure 4: Signal-imbalanced QAT dynamics. The weak signal at q_1 causes slow drift toward the boundary, whereas the larger signal at q_2 quickly pulls the shadow back.

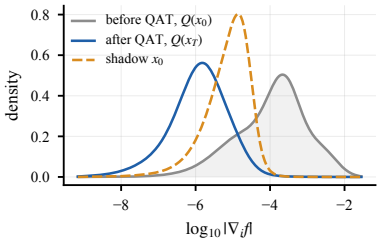


Figure 5: Llama-2-7B 2-bit: distribution of $\log_{10}|\nabla_i f|$ before QAT ($Q(\mathbf{x}_0)$, RTN), after QAT, and at the pretrained shadow $\mathbf{x}_0$.

Stochastic gradients can further reinforce this asymmetry later in training, when gradient signals become small relative to mini-batch noise. At a weak-signal endpoint such as q_1, noise can flip the effective update direction and undo part of the accumulated motion toward the boundary, further lengthening the residence time. Once the shadow crosses to q_2, the stronger opposing signal is less easily reversed by noise and quickly drives it back toward q_1. This history-dependent cycle biases how long the shadow remains at each endpoint and which crossings persist, even when the pooled crossing rate is well predicted.

Algorithm 1 QAR framework: one endpoint step on the finite codebook

Input: code $\mathbf{k}_t \in \mathcal{K} = \mathcal{K}_1 \times \cdots \times \mathcal{K}_d$, spacings $\mathbf{s}$, stepsize η, route map $\mathcal{U}_t$, amplifier map $\mathcal{A}$
1: recover endpoint $\mathbf{q}_t = \mathbf{s} \odot \mathbf{k}_t$
2: compute endpoint-gradient estimate $\widehat{\mathbf{g}}_t$ at $\mathbf{q}_t$
3: set route $\mathbf{u}_t = \mathcal{U}_t(\widehat{\mathbf{g}}_t)$
4: set direction $\mathbf{z}_t = \mathrm{sign}(\mathbf{u}_t)$
5: set feasibility mask $\mathbf{M}_t = \mathbf{1}_{\mathcal{K}}(\mathbf{k}_t - \mathbf{z}_t)$
6: set probability $\mathbf{p}_t = \min\{\mathcal{A}(\eta, \mathbf{u}_t, \mathbf{s}, \mathbf{M}_t), \mathbf{1}\}$
7: sample $\mathbf{J}_t \sim \mathrm{Bernoulli}(\mathbf{p}_t)$ coordinatewise
8: set $\mathbf{k}_{t+1} = \mathbf{k}_t - \mathbf{z}_t \odot \mathbf{J}_t$
Output: updated code $\mathbf{k}_{t+1}$

Consistent with the asymmetric dynamics, the endpoint gradient distribution after 2-bit QAT on Llama-2-7B shifts toward smaller magnitudes, with a substantially reduced large-gradient tail (Figure 5). This motivates a direct endpoint update that prioritizes transitions associated with stronger signals.

4.2 The Proposed Algorithm

The crossing law of Section 3 turns shadow updates into probabilistic one-cell endpoint transitions, and Section 4.1 suggests assigning more of this transition probability to coordinates with stronger signals. We therefore propose **QAR** (**Q**uantization with **A**mplified **R**outing), which samples these transitions directly on the integer code rather than producing them through a shadow weight. A route map selects the update signal, and an amplifier map converts it into coordinatewise jump intensities.

Formally, let

$$\mathcal{K}_i = \{\underline{k}_i, \underline{k}_i + 1, \ldots, \overline{k}_i\} \subset \mathbb{Z}, \quad \mathcal{Q} = \{\mathbf{q} : \; q_i = s_i k_i, \; k_i \in \mathcal{K}_i, \; i = 1, \ldots, d\},$$

where $\underline{k}_i \le \overline{k}_i$ are integers; the symmetric b-bit codebook is the special case $\mathcal{K}_i = \mathcal{K}_b$. For a candidate code $\mathbf{v}$, define the feasibility indicator $\mathbf{1}_{\mathcal{K}}(\mathbf{v})_i = \mathbf{1}\{v_i \in \mathcal{K}_i\}$. Coordinates are partitioned into scale-sharing groups G, with $s_i = s_G$ for $i \in G$.

The direction $\mathbf{z}_t = \mathrm{sign}(\mathbf{u}_t)$ is kept separate from the amplifier $\mathcal{A} : \mathbb{R}_+ \times \mathbb{R}^d \times \mathbb{R}^d_{++} \times \{0,1\}^d \to \mathbb{R}^d_+$, which returns nonnegative intensities and, through $\mathbf{M}_t$, assigns zero intensity to moves that would leave the codebook. Hence $\mathbf{k}_{t+1} \in \mathcal{K}$ whenever $\mathbf{k}_t \in \mathcal{K}$. The minimum in Algorithm 1 is taken coordinatewise. Because QAR updates $\mathbf{k}_t$ directly, its persistent weight state is $\mathbf{k}_t$ plus one scale s_G per group, and $\mathbf{q}_t = \mathbf{s} \odot \mathbf{k}_t$ is materialized only for computation. It therefore stores less than QAT, which keeps a full-precision shadow copy (Appendix B.5).

Uniform amplifier. Reproducing Model 2 on feasible coordinates gives

$$\mathcal{A}_{\mathrm{unif}}(\eta, \mathbf{u}, \mathbf{s}, \mathbf{M}) = \eta \, \mathbf{M} \odot |\mathbf{u}| \oslash \mathbf{s}, \tag{7}$$

where $\oslash$ denotes elementwise division. Away from the codebook boundary and whenever $\eta|u_{t,i}| \le s_i$, QAR with $\mathcal{A}_{\mathrm{unif}}$ follows the transition law of Model 2 exactly. With the SGD route $\mathbf{u}_t = \widehat{\mathbf{g}}_t$, it coincides with the DQT stochastic-rounding update (Zhao et al., 2025) on a fixed uniform codebook, including its boundary clipping; Appendix A compares QAR with DQT and other direct low-precision methods.

Power amplifier. The uniform amplifier treats all coordinates in a group alike. To favor stronger signals, let $a_i = M_i|u_i|$ be the feasible update magnitude and define the normalized power score, for $\gamma \ge 0$,

$$b_{\gamma,i} = \begin{cases} \frac{a_i^\gamma}{|G|^{-1} \sum_{j \in G} a_j^\gamma}, & \text{if } \gamma > 0 \text{ and } \sum_{j \in G} a_j^\gamma > 0, \\ 1, & \text{else,} \end{cases} \qquad i \in G. \tag{8}$$

The power amplifier rescales the uniform one by this score:

$$\mathcal{A}_\gamma = \mathcal{A}_{\mathrm{unif}} \odot \mathbf{b}_\gamma, \qquad \text{i.e.} \qquad \mathcal{A}_{\gamma,i}(\eta, \mathbf{u}, \mathbf{s}, \mathbf{M}) = \frac{\eta a_i}{s_G} \, b_{\gamma,i}, \quad i \in G. \tag{9}$$

We refer to Algorithm 1 instantiated with $\mathcal{A}_\gamma$ as QAR-γ. Since $b_{0,i} = 1$, QAR-0 is the uniform-amplifier update. For $\gamma > 0$, $\mathbf{b}_\gamma$ has mean one within each group, so it reweights coordinates toward larger update magnitudes rather than acting as a global stepsize multiplier.

4.3 Finite-Grid Feasible-Gradient Bounds for QAR

At a codebook boundary, the full gradient need not vanish because one descent direction can be infeasible. For $q_i = s_i k_i$, define the feasible endpoint gradient by

$$\big(\mathbf{g}_{\mathcal{Q}}(\mathbf{q})\big)_i = g_i \, \mathbf{1}\{k_i - \operatorname{sign}(g_i) \in \mathcal{K}_i\}. \tag{10}$$

It retains exactly the components with a feasible one-cell descent move and equals the full gradient away from the boundary.

Theorem 4 (Feasible-gradient bound for QAR-γ)**.** *Suppose Assumptions 1 and 2 hold. Write $f^*_{\mathcal{Q}} = \min_{\mathbf{q}\in\mathcal{Q}} f(\mathbf{q})$. For each group G, let $L_G = \max_{i\in G} L_i$. For any $\gamma \geq 0$, run QAR-γ according to Algorithm 1, and let $b_{t,i}$ denote the power score of coordinate i at step t. Assume almost surely that $\eta U b_{t,i} \leq s_G$ for every $t < T$ and $i \in G$.*

If $\mathbf{u}_t = \nabla f(\mathbf{q}_t)$ (GD), then

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\big[\|\mathbf{g}_{\mathcal{Q}}(\mathbf{q}_t)\|_2^2\big] \leq \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\sum_G \sum_{i\in G} b_{t,i}\big(\mathbf{g}_{\mathcal{Q}}(\mathbf{q}_t)\big)_i^2\right] \leq \frac{2(f(\mathbf{q}_0) - f^*_{\mathcal{Q}})}{\eta T} + \frac{1}{4}\sum_G |G| L_G^2 s_G^2. \tag{11}$$

If $\mathbf{u}_t = \operatorname{sign}(\nabla f(\mathbf{q}_t))$ (signGD), then

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\|\mathbf{g}_{\mathcal{Q}}(\mathbf{q}_t)\|_1] \leq \frac{f(\mathbf{q}_0) - f^*_{\mathcal{Q}}}{\eta T} + \frac{1}{2}\sum_G |G| L_G s_G. \tag{12}$$

Remark 1 (Effect of signal imbalance). The first inequality in (11) follows from Lemma 7. It is an equality for $\gamma = 0$ and is strict for $\gamma > 0$ whenever the feasible-gradient magnitudes are unequal within at least one group. This shows how leveraging signal imbalance yields a strictly stronger stationarity certificate than uniform routing.

For comparison, let $\mathbf{y}_{t+1} = \mathbf{y}_t - \eta \nabla f(\mathbf{y}_t)$. For an L-smooth objective, classical nonconvex GD with fixed $0 < \eta \leq 1/L$ satisfies

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla f(\mathbf{y}_t)\|_2^2 \leq \frac{2(f(\mathbf{y}_0) - \inf_{\mathbf{y}} f(\mathbf{y}))}{\eta T} = O(T^{-1}).$$

Equation (11) replaces the full gradient by the feasible endpoint gradient and adds a grid floor, while $\eta U b_{t,i} \leq s_G$ ensures that the transition probabilities are not truncated. For gradient update, the bound of QAR decreases at rate $O(T^{-1})$ until it reaches the fixed-grid floor $\frac{1}{4}\sum_G |G| L_G^2 s_G^2$. The sign update instead controls the average feasible ℓ_1 gradient by $O(T^{-1})$ plus its grid floor. Both finite-grid floors are unavoidable under the stated assumptions. The proof and a construction showing that these finite-grid floors are tight appear in Appendix E. The stochastic extension is deferred to Appendix F, where we further characterize the tradeoff between exploiting signal imbalance and amplifying stochastic-gradient noise.

4.4 Experiments

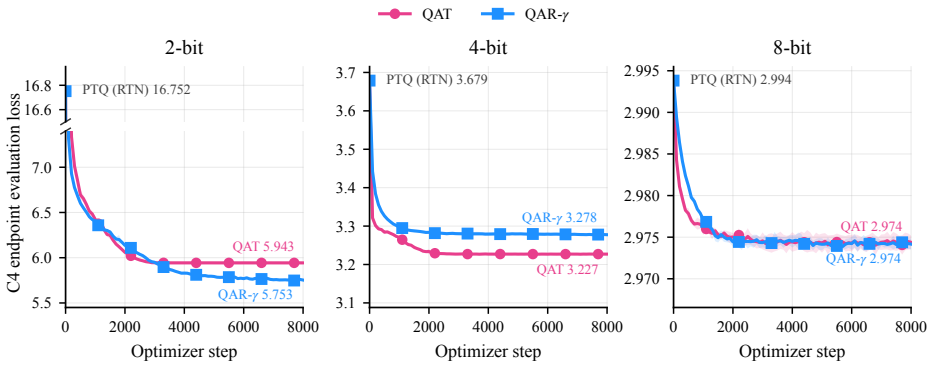


Figure 6: Qwen-2.5-0.5B C4 endpoint-loss curves for the best settings of QAT and QAR-γ at 2, 4, and 8 bits. Curves show the mean over three seeds, and shading shows ± 1 sample standard deviation.

Table 1: Llama-3.2-3B and Llama-2-7B results. PTQ (RTN) is the shared initialization of QAT and QAR-γ.

Model	Bits	Method	C4 PPL ↓	Wiki PPL ↓	HellaSwag	PIQA	Wino.	ARC-E	ARC-C	OBQA	BoolQ	LAMBADA	MMLU	Avg. ↑
Llama-3.2-3B	16	FP (BF16)	13.47	11.86	73.67	77.42	69.93	71.68	46.08	43.60	73.06	70.13	53.95	64.39
	4	PTQ (RTN)	20.67	18.21	66.81	74.70	62.35	62.25	37.37	36.40	68.10	57.89	42.10	56.44
		QAT	**15.06**	**13.71**	70.41	77.15	65.67	69.11	42.32	38.00	72.05	64.84	49.49	**61.00**
		QAR-γ	**15.04**	**13.75**	70.43	76.99	65.04	69.28	42.15	37.60	72.17	64.84	49.49	**60.89**
	8	PTQ (RTN)	13.27	11.93	73.57	77.31	69.22	71.38	46.25	43.20	72.87	69.98	54.26	64.23
		QAT	**12.33**	**11.20**	73.85	77.48	69.14	72.18	46.16	42.60	73.58	69.78	54.72	**64.39**
		QAR-γ	**12.32**	11.48	73.91	77.58	69.46	72.52	46.42	42.20	73.27	69.55	54.71	**64.40**
Llama-2-7B	16	FP (BF16)	8.19	7.22	75.97	78.73	69.06	74.87	46.16	43.80	77.86	73.47	40.69	64.51
	4	PTQ (RTN)	20.61	18.94	59.94	74.16	65.67	68.01	42.49	40.00	69.20	61.21	29.96	56.74
		QAT	10.07	9.01	74.37	76.39	66.61	70.37	44.62	42.00	71.71	61.40	32.50	60.00
		QAR-γ	**9.28**	**8.42**	74.49	77.97	68.90	70.20	44.03	42.00	71.77	66.76	34.88	**61.22**
	8	PTQ (RTN)	8.19	7.23	75.96	78.84	68.35	74.58	45.56	44.00	77.68	73.90	40.61	64.39
		QAT	**7.79**	**7.07**	77.25	79.16	69.06	75.25	46.33	44.80	78.84	72.64	41.03	**64.93**
		QAR-γ	**7.77**	**7.05**	76.98	78.67	70.01	74.45	45.82	44.40	78.32	73.34	41.06	**64.78**

We compare QAT and QAR-γ in post-training of Qwen-2.5-0.5B, Llama-3.2-3B, and Llama-2-7B on fixed per-output-channel weight grids; Appendix B gives the training and evaluation settings. On Qwen-2.5-0.5B, QAR-γ has lower C4 endpoint loss than QAT at 2 bits and remains competitive with QAT at 4 and 8 bits (Figure 6). On Llama-3.2-3B, QAR-γ matches QAT at 4 and 8 bits. On Llama-2-7B, QAR-γ lowers C4 perplexity from 10.07 to 9.28 and raises average downstream accuracy from 60.00 to 61.22 at 4 bits, and is close to QAT at 8 bits (Table 1). Table 2 further shows that QAR uses less peak CUDA memory at both reported bit widths, with savings of up to 24.1% that grow as the bit width decreases.

Table 2: Peak CUDA memory cost (GiB) for QAT and QAR-γ. Parentheses show reductions relative to QAT.

Model	QAT	QAR-γ	
		8-bit	4-bit
Llama-3.2-3B	17.33	14.75 (14.9%)	13.43 (22.5%)
Llama-2-7B	37.43	31.44 (16.0%)	28.42 (**24.1%**)

We also varied the QAR power exponent γ and the initial learning rate on Qwen-2.5-0.5B. Amplification improves the 2-bit and 4-bit results over uniform routing, and QAR's final loss varies less across the tested learning rates than QAT's (Appendix B.4). QAR's advantages in quality, memory, and learning-rate robustness are clearest under aggressive quantization. We also tested ECO (Nikdan et al., 2026), which removes full-precision master weights through momentum error-feedback, under the same Qwen-2.5-0.5B post-training protocol; Appendix B.6 gives the results and diagnostics.

5 Conclusion and Limitations

QAT updates a full-precision shadow weight but deploys a quantized endpoint, and the relevant object of analysis is therefore the deployed endpoint trajectory. We show how homogeneous residual model turns endpoint motion into a probabilistic transition on the finite grid and how signal imbalance can create asymmetric oscillations, motivating QAR's update on the quantization code without keeping a full-precision shadow weight. We derive feasible-gradient bounds for QAR with power amplifier and discuss the tradeoff between the gain from signal imbalance and the penalty of strengthening stochastic noise (see Appendix F). Experiments support the endpoint view, especially in low-bit LLM post-training, and show that QAR is comparable to or better than QAT at lower peak memory.

Limitations. This work focuses on weight-only quantization. In this setting, the deployed model lives on a finite weight grid. Our analysis tracks how STE updates move the shadow weights across the cells of this grid. Practical low-bit QAT often quantizes activations as well, where the quantized values depend on the input data and the endpoint can change through both cell crossings and activation-level changes. Extending the finite-grid endpoint view to joint weight-activation QAT is an important direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools to help refine the proofs of the stochastic QAR guarantees in Appendix F. Starting from an initial bound derived by the authors, the AI tools suggested (i) splitting the SGD-route analysis of Theorem 10 into two cases, depending on whether the true descent direction is feasible or blocked at a boundary, which improved the constant and removed a redundant noise-floor term $\langle \mathbf{L} \odot \mathbf{s}, \boldsymbol{\sigma} \rangle / \sqrt{n}$; (ii) using Cantelli's one-sided inequality to bound the wrong-sign probability in the signSGD route; and (iii) a simpler proof of Lemma 12. The authors independently checked every AI-suggested proof step for mathematical correctness and consistency with the stated assumptions.

Additionally, we used generative AI tools to identify candidate related work, to refine the wording throughout the paper, and to help implement the large language model experiments. We have not used generative AI tools for research ideation, drafting sections of the paper or generating synthetic datasets. The authors reviewed and tested the AI-assisted experiment code, manually verified suggested related work against the original sources, and reviewed and edited all AI-assisted wording to preserve the intended technical meaning and claims. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

For the theoretical results, the assumptions are stated explicitly in Section 2 (Assumptions 1 and 2), Section 3, and Appendix F (Assumption 3), and complete proofs of all claims are given in Appendices D, E, and F. The QAR update is specified in Algorithm 1. For the experiments, Table B.1 and Appendix B.1 list the models, quantizer and scale selection, training scope, data, optimizer settings, learning-rate schedules, batch sizes, seeds, and hardware; the learning-rate and power-exponent selection sweeps are reported in Appendix B.4, and the memory measurement protocol in Appendix B.5. All models and datasets used are publicly available.

ETHICS STATEMENT

This work is methodological and does not involve human-subject research, the collection of personal data, or the release of a new dataset. Our experiments use publicly available pretrained models and datasets solely to evaluate optimization behavior, predictive performance, and memory use. By reducing the memory required for low-bit model training, QAR may broaden access to model adaptation and deployment.

REFERENCES

Yu Bai, Yu-Xiang Wang, and Edo Liberty. ProxQuant: Quantized neural networks via proximal operators. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

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

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signSGD: Compressed optimisation for non-convex problems. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 560–569, 2018.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 1877–1901, 2020.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. EfficientQAT: Efficient quantization-aware training for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10081–10100. Association for Computational Linguistics, 2025.

Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. PACT: Parameterized clipping activation for quantized neural networks. *arXiv preprint*, arXiv:1805.06085, 2018.

Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. BinaryConnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems 28 (NeurIPS)*, 2015.

Valentin De Bortoli, Alain Durmus, Xavier Fontaine, and Umut Şimşekli. Quantitative propagation of chaos for SGD in wide neural networks. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 278–288, 2020.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, pp. 10088–10115, 2023.

Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Koen Helwegen, James Widdicombe, Lukas Geiger, Zechun Liu, Kwang-Ting Cheng, and Roeland Nusselder. Latent weights do not exist: Rethinking binarized neural network optimization. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pp. 7531–7542, 2019.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pp. 30016–30030, 2022.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2704–2713, 2018.

Saqib Javed, Hieu Le, and Mathieu Salzmann. QT-DoG: Quantization-aware training for domain generalization. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pp. 26981–27004, 2025.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

Taejong Joo, Wenhan Xia, Cheolmin Kim, Ming Zhang, and Eugene Ie. On surprising effectiveness of masking updates in adaptive optimizers. *arXiv preprint*, arXiv:2602.15322, 2026.

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.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

Junghyup Lee, Dohyung Kim, and Bumsub Ham. Network quantization with element-wise gradient scaling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6448–6457, 2021.

Junghyup Lee, Jeimin Jeon, Dohyung Kim, and Bumsub Ham. Scheduling weight transitions for quantization-aware training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23466–23475, 2025.

Guoqi Li, Lei Deng, Lei Tian, Haotian Cui, Wentao Han, Jing Pei, and Luping Shi. Training deep neural networks with discrete state transition. *Neurocomputing*, 272:154–162, 2018.

Hanyang Li, Jianhao Ma, and Ying Cui. Understanding quantization-aware training: Gradients at quantized weights bias to the low-loss basin. *arXiv preprint*, arXiv:2606.09012, 2026.

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

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. LLM-QAT: Data-free quantization aware training for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 467–484. Association for Computational Linguistics, 2024.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

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

Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. Overcoming oscillations in quantization-aware training. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pp. 16318–16330, 2022.

Mahdi Nikdan, Amir Zandieh, Dan Alistarh, and Vahab Mirrokni. ECO: Quantized training without full-precision master weights. *arXiv preprint*, arXiv:2601.22101, 2026.

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

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks: A law of large numbers. *SIAM Journal on Applied Mathematics*, 80(2):725–752, 2020.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R. Steeves, Joel Hestness, and Nolan Dey. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. https://cerebras.ai/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama, 2023.

Kaiyue Wen, Zhiyuan Li, Jason Wang, David Hall, Percy Liang, and Tengyu Ma. Understanding warmup-stable-decay learning rates: A river valley loss landscape view. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, pp. 42840–42885, 2025.

Jonathan Wenshøj, Bob Pepin, and Raghavendra Selvan. Oscillations make neural networks robust to quantization. *Transactions on Machine Learning Research*, 2025.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 38087–38099, 2023.

Kaiyan Zhao, Tsuguchika Tabaru, Kenichi Kobayashi, Takumi Honda, Masafumi Yamazaki, and Yoshimasa Tsuruoka. Direct quantized training of language models with stochastic rounding. In *Proceedings of the 17th Asian Conference on Machine Learning (ACML)*, pp. 1150–1165, 2025.

Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint*, arXiv:1606.06160, 2016.

APPENDIX A RELATED WORK

Direct low-precision weight updates. Direct updates of discrete weights predate recent low-bit language-model training. Discrete State Transition (DST) uses probabilistic projection to keep weights in a configurable multilevel discrete space (Li et al., 2018). In the one-cell regime, DST's nonlinear projection can be represented within QAR by a tanh amplifier. Bop instead stores binary weights and an exponential moving average of their gradients, then deterministically flips a weight when the accumulated signal exceeds a threshold and points in the flipping direction (Helwegen et al., 2019). Direct Quantized Training (DQT) applies stochastic rounding to a tentative optimizer update on a low-bit language-model weight grid (Zhao et al., 2025), which QAR-0 recovers as a special case (Section 4.2). ECO (Nikdan et al., 2026) also removes persistent master weights by injecting the quantization error of a temporary update into optimizer momentum. QAR approaches direct discrete updates from a different starting point: it derives a transition law from QAT's boundary-crossing dynamics and factors that law into a route and an amplifier. This formulation supports signal-dependent power amplification and a finite-codebook analysis in terms of feasible endpoint gradients.

Transition-rate scheduling. One way to regulate endpoint motion in QAT is to control how often shadow weights cross quantization boundaries. Lee et al. (2025) adapt the shadow-weight learning rate to track a prescribed fraction of quantized weights that change levels. QAR instead samples coordinatewise endpoint transitions directly.

Update scaling. Boundary crossings can also be influenced by changing the magnitude or frequency of shadow-weight updates. Lee et al. (2021) use the gradient sign and quantization residual to rescale each STE update, with a Hessian-based rule for adapting the scaling factor. Joo et al. (2026) randomly mask blockwise optimizer updates while retaining dense moment updates and use momentum–gradient alignment to control the masking probability. These approaches modify shadow-weight updates, whereas QAR leverages signal imbalance to assign larger transition probabilities to feasible one-cell endpoint updates with stronger gradient signals.

APPENDIX B ADDITIONAL EXPERIMENTAL DETAILS AND DIAGNOSTICS

Table B.1: Experimental settings for the Qwen-2.5-0.5B, Llama-3.2-3B, and Llama-2-7B runs

setting	Qwen-2.5-0.5B	Llama-3.2-3B	Llama-2-7B
compute resource	NVIDIA A40	NVIDIA A100	NVIDIA H100
sequence length	512	512	1024
quantizer	fixed symmetric per-output-channel uniform weight quantization, initialized by RTN		
bit widths	$2, 4, 8$	$4, 8$	$4, 8$
scale selection	$s_c = \max_j \lvert W_{cj}\rvert/(2^{b-1}-1)$, fixed during training		
training scope	quantized linear weights only; biases, embeddings, normalizations, and all other parameters frozen		
training data	tokens from SlimPajama-627B (Soboleva et al., 2023)		
batch size	1	4	4
seeds	$0, 1, 2$	0	0
evaluation data	C4 validation (Raffel et al., 2020)	C4 and WikiText-2 (Merity et al., 2017) perplexity; nine zero-shot tasks	
optimizer	AdamW with QAT $(\beta_1, \beta_2) = (0.9, 0.95)$ and QAR-γ $(\beta_1, \beta_2) = (0.95, 0.9)$, $\epsilon = 10^{-8}$		
weight decay	0		
gradient clipping	0 (disabled)		
learning-rate schedule	no warmup; cosine decay to $0.1\eta_0$ over the first half of the training budget, then held fixed		

B.1 EXPERIMENTAL SETTINGS

We use Qwen-2.5-0.5B, Llama-3.2-3B, and Llama-2-7B to evaluate the finite-grid predictions on post-training QAT runs and compare QAT with QAR-γ. The Qwen runs use bit widths $b \in \{2, 4, 8\}$, while the Llama runs use $b \in \{4, 8\}$. For each model, both methods start from the same pretrained checkpoint. We quantize linear-layer weights except the language-model head using fixed symmetric per-output-channel uniform quantization initialized by round-to-nearest (RTN). For each output channel c, we set $s_c = \max_j |W_{cj}|/q_{\max}$, where W_{cj} is the weight in output channel c and input coordinate j, and $q_{\max} = 2^{b-1} - 1$, then round and clip the integer levels to $[-q_{\max}, q_{\max}]$. The

scales remain fixed for both methods. We use the same quantizer family at every bit width so that bit width is the only factor that varies across the comparisons.

We train all three models on SlimPajama-627B (Soboleva et al., 2023) and use C4 validation (Raffel et al., 2020) to evaluate the endpoint. For both Llama models, we additionally report WikiText-2 perplexity (Merity et al., 2017) and nine zero-shot tasks. All runs use gradient accumulation 1. The Qwen-2.5-0.5B and Llama-3.2-3B runs use sequence length 512 on an NVIDIA A40 and A100, respectively; the Llama-2-7B runs use sequence length 1024 on an NVIDIA H100. The Qwen runs use minibatch size 1, whereas both Llama models use minibatch size 4. We use an 8,000-step budget for Qwen-2.5-0.5B and a 10,000-step budget for Llama-3.2-3B. On Llama-2-7B, we find that QAT easily drifts away from its best checkpoint as the training horizon increases, so we use 4,000 steps at 4 bits and 2,000 steps at 8 bits.

All runs use AdamW with QAT $(\beta_1, \beta_2) = (0.9, 0.95)$ and QAR-γ $(\beta_1, \beta_2) = (0.95, 0.9)$, $\epsilon = 10^{-8}$, zero weight decay, and no gradient clipping. The learning rate decays without warmup to $0.1\eta_0$ over the first half of the training budget and remains fixed thereafter.

B.2 Endpoint and Gradient Diagnostics

Endpoint diagnostics on loss trajectory and landscape. Figure 7 tracks both losses at each bit width's best learning rate: at 2 and 4 bits the C4 endpoint loss decreases while the shadow loss increases substantially. Figure 8 shows the same reversal in the loss landscape at the initial and final checkpoints.

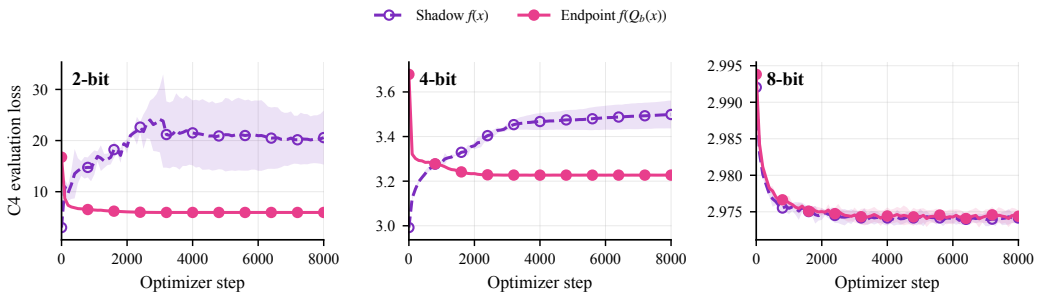


Figure 7: Qwen-2.5-0.5B QAT shadow loss $f(x)$ versus deployed-endpoint loss $f(Q_b(x))$. Curves show the mean over three seeds; shading shows ± 1 sample standard deviation.

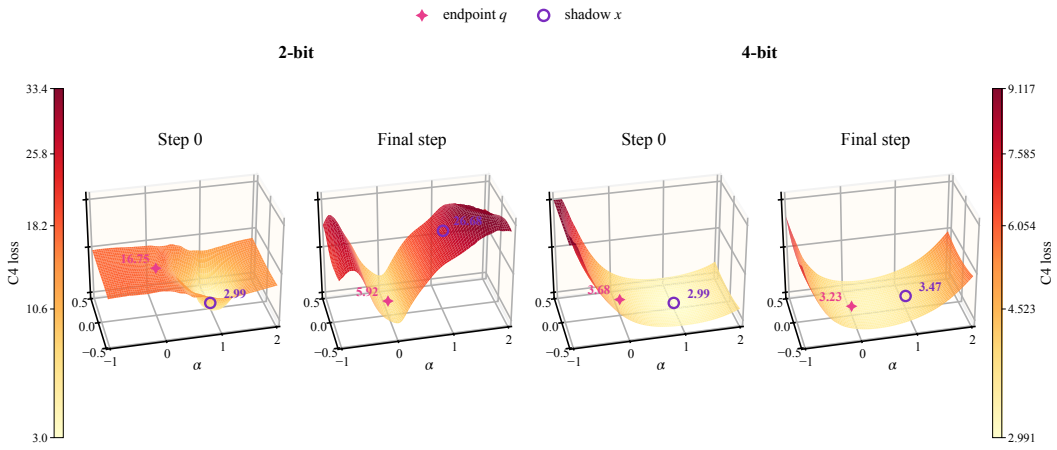


Figure 8: Qwen-2.5-0.5B C4 evaluation-loss landscape at step 0 and at final step for seed 0. The surface evaluates $Q_b(\mathbf{x}) + \alpha(\mathbf{x} - Q_b(\mathbf{x})) + \beta\|\mathbf{x} - Q_b(\mathbf{x})\|_2\mathbf{v}$ for a random unit $\mathbf{v} \perp (\mathbf{x} - Q_b(\mathbf{x}))$.

Gradient diagnostics. Figure 9 tracks the average magnitude of active gradient coordinates, $\|\mathbf{g}\|_1/\|\operatorname{sign}(\mathbf{g})\|_1$. At the quantized endpoint, both QAT and QAR-γ reduce this quantity, most

clearly at 2 bits, showing that both methods remove strong endpoint-gradient signals. In contrast, QAT's shadow-space value can remain much larger at low precision, reinforcing that the shadow trajectory is not a faithful diagnostic of endpoint progress.

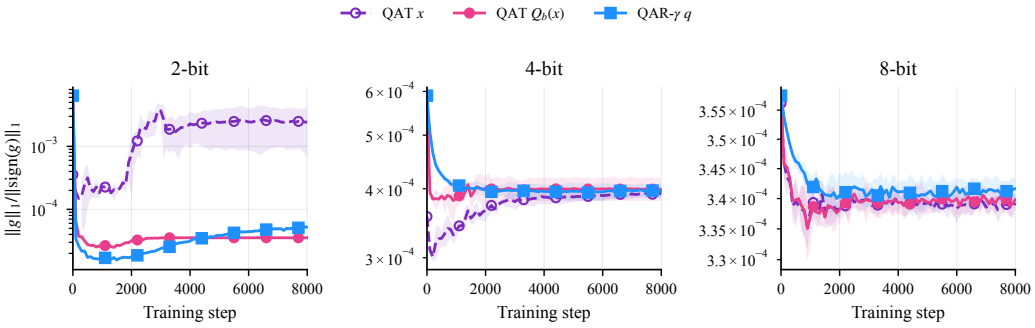


Figure 9: Qwen-2.5-0.5B active-gradient ratio at 2, 4, and 8 bits: QAT at x, QAT at $Q_b(x)$, and QAR-γ at q, with $\gamma = (16, 2, 0)$, respectively. Curves show the mean over three seeds; shading shows ± 1 sample standard deviation.

B.3 Crossing-Rate Diagnostics

For the realized shadow step from $\mathbf{x}_t$ to $\mathbf{x}_{t+1}$, let $\mathcal{I}_t$ contain the active non-saturated coordinates and define the predicted and observed aggregate crossing rates by

$$\widehat{p}_t = \frac{1}{|\mathcal{I}_t|} \sum_{i \in \mathcal{I}_t} \min\left\{ \frac{|(\mathbf{x}_{t+1})_i - (\mathbf{x}_t)_i|}{s_i}, 1 \right\}, \qquad \widehat{c}_t = \frac{1}{|\mathcal{I}_t|} \sum_{i \in \mathcal{I}_t} \mathbf{1}\{Q_b(\mathbf{x}_{t+1})_i \neq Q_b(\mathbf{x}_t)_i\}.$$

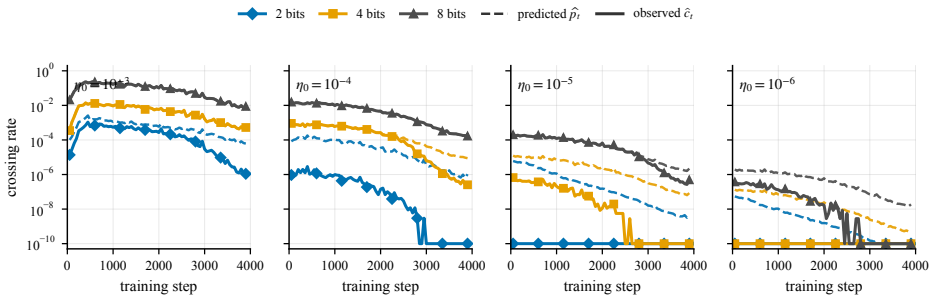


Figure 10: QAT crossing rates on Qwen-2.5-0.5B across four learning rates and 2, 4, and 8 bits. Dashed curves show the prediction $\widehat{p}_t$ and solid curves show the observed rate $\widehat{c}_t$. Zero observations are displayed at 10^{-10}.

Figures 10 and 11 compare predicted crossing rates with realized fixed-grid crossings. Figure 10 sweeps $\eta_0 \in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$ for QAT, while Figure 11 shows the best QAT and QAR-γ settings at 2, 4, and 8 bits; for QAR, the prediction is its planned probability $\mathbf{p}_t$.

In both figures, the realized QAT rate collapses late in training while its prediction stays nonzero. This collapse comes from BF16 rounding rather than a failure of the residual-phase model. BF16 values are not evenly spaced, and the gap between adjacent values grows with their magnitude. When an update is smaller than half the local gap around x_i, round-to-nearest maps $x_i + \Delta x_i$ back to x_i, so the shadow weight neither moves nor crosses a quantization boundary. More coordinates enter this regime as the learning rate decays. Figure 12 repeats the diagnostic with FP32 shadow weights to separate this numerical effect from residual-phase calibration.

With FP32, the late-training collapse disappears. At 4 and 8 bits, the realized-to-predicted ratios stay near one or modestly above it. Their gradual upward drift after the initial transient is consistent with the toy-model analysis of Wenshøj et al. (2025), in which a residual-driven STE component pushes shadow weights away from quantization levels and toward cell boundaries. This boundary

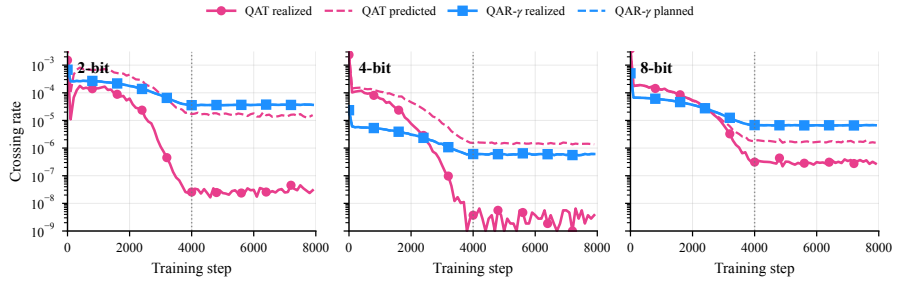


Figure 11: Predicted versus realized crossing rates for the best QAT and QAR-γ settings; for QAR, the prediction is the planned probability. Dashed lines are predicted; solid lines are realized.

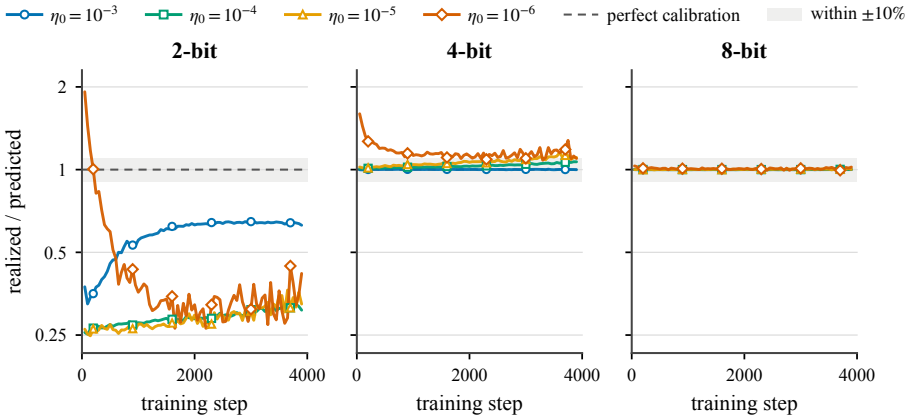


Figure 12: FP32 control for Figure 10. Curves show the ratio of the realized to the predicted boundary-crossing rate for each initial learning rate; the dashed line marks perfect calibration and the shaded band marks $\pm 10\%$.

concentration puts more residual mass in the crossing region than the uniform-phase model predicts, causing the realized crossing rate to exceed the prediction. The 2-bit ratios remain below one, consistent with the center-concentrated phase of its single non-saturated interior cell.

At $\eta_0 = 10^{-6}$, Figure 12 shows an initial spike because the run starts from a BF16 checkpoint. BF16 stores weights on a discrete lattice, placing about 0.1%–0.2% of them exactly on quantization boundaries. Even a tiny first update can move these weights across a boundary, producing more crossings than the continuous phase model predicts. The spike fades once training moves the weights off the BF16 lattice.

QAR has no shadow weight to round, so its realized rate tracks its planned rate throughout training (Figure 11).

B.4 Ablation Studies on QAR

Power-exponent ablation. Figure 13 sweeps $\gamma \in \{0, 1, 2, 4, 8, 16, 32\}$ for Qwen-2.5-0.5B. At 2 bits, final C4 loss falls substantially between the uniform-routing choice $\gamma = 0$ and the minimum at $\gamma = 16$. The minimum occurs at $\gamma = 2$ at 4 bits, with a much smaller difference from $\gamma = 0$. At 8 bits, $\gamma = 0$ gives the lowest loss. Thus stronger signal concentration is useful on the coarsest grid, but its value disappears as the grid becomes finer, consistent with the signal–noise tradeoff in Appendix F. Since the one-cell condition $\eta|u_{t,i}| \leq s_i$ holds at every step, the $\gamma = 0$ arm is exactly DQT (Section 4.2), so the sweep also serves as a DQT comparison.

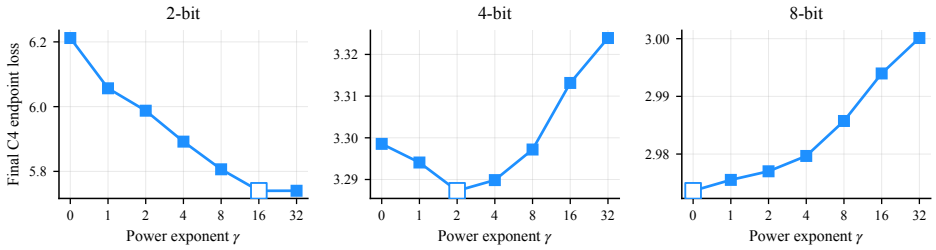


Figure 13: Final C4 endpoint loss after 8000 steps of Qwen-2.5-0.5B post-training on fixed per-output-channel weight grids (seed 0), plotted against the QAR power exponent. Hollow markers identify the lowest loss among the displayed exponents at each bit width.

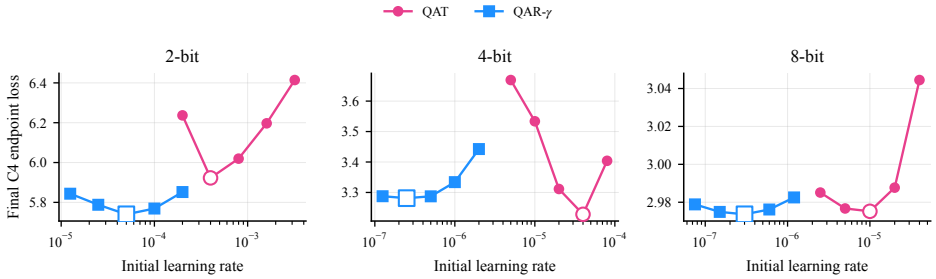


Figure 14: Same setting as Figure 13, plotted against the initial learning rate. QAR uses $\gamma = 16, 2, 0$ at 2, 4, and 8 bits, respectively. Hollow markers identify the lowest loss on each displayed method curve.

Learning-rate sensitivity. Figure 14 plots seed-0 final C4 loss over the tested initial learning rates. The lowest-loss QAR-γ settings use smaller initial rates than the lowest-loss QAT settings. Across the displayed rates, the QAR-γ curves also vary less in final loss.

B.5 Memory Diagnostic

Persistent state. Let P be the total number of model parameters, N the number of quantized coordinates updated by both methods, and R the number of quantization groups. Both methods freeze the remaining $P - N$ parameters. If a shadow value, optimizer moment, and group scale each use w, m, and r bytes, then

$$M_{\mathrm{QAT}} = w(P - N) + wN + 2mN + rR, \qquad M_{\mathrm{QAR}} = w(P - N) + \frac{b}{8}N + 2mN + rR,$$

where QAR-γ stores b-bit endpoint codes instead of a shadow tensor. Its persistent-state saving is therefore $(w - b/8)N$ bytes.

Table B.2: Numbers of total parameters P, quantized parameters N, and quantization groups R, and the resulting persistent-state savings of QAR-γ relative to QAT with BF16 weights and moments and FP32 group scales.

Model	P	N	R	Persistent-state saving		
				2-bit	4-bit	8-bit
Qwen-2.5-0.5B	494,032,768	357,826,560	304,128	25.9%	22.2%	14.8%
Llama-3.2-3B	3,212,749,824	2,818,572,386	774,144	27.9%	23.9%	15.9%
Llama-2-7B	6,738,415,616	6,476,005,376	1,359,872	28.8%	24.7%	16.4%

With BF16 weights and moments and FP32 group scales, the two totals are $2(P - N) + 6N + 4R$ and $2(P - N) + (4 + b/8)N + 4R$. Our implementation stores 2- and 4-bit endpoint codes in packed form and updates them with fused packed CUDA kernels. Table B.2 lists the resulting persistent-state savings.

Peak memory. Peak allocation also depends on transient gradient buffers. In our implementation, we adopt a streamed update in which QAR consumes and releases each layer gradient during backward, while its fused sampler avoids model-shaped temporaries. For a fair comparison, we additionally stream the QAT AdamW update in the same way. Peak memory in Table 2 is measured on an NVIDIA A40 with sequence length 512, batch size 1, and gradient checkpointing, as the median over 20 steps after 5 warm-up steps.

B.6 Comparison with ECO

We compare ECO with QAT and QAR-γ under the Qwen-2.5-0.5B protocol of Appendix B.1, with 8000 training steps at 2, 4, and 8 bits. ECO (Nikdan et al., 2026) removes the full-precision copy by injecting the quantization error of a tentative AdamW step into its first moment. In our notation, one ECO step is

$$\begin{aligned}
\tilde{\mathbf{m}}_{t+1} &= \beta_1 \hat{\mathbf{m}}_t + (1-\beta_1)\nabla f(\mathbf{q}_t), \\
\tilde{\mathbf{x}}_{t+1} &= \mathbf{q}_t - \eta_t \mathbf{D}_t^{-1} \tilde{\mathbf{m}}_{t+1}, \\
\mathbf{e}_{t+1} &= \tilde{\mathbf{x}}_{t+1} - Q_b(\tilde{\mathbf{x}}_{t+1}), \\
\hat{\mathbf{m}}_{t+1} &= \tilde{\mathbf{m}}_{t+1} + \underbrace{\lambda \frac{1}{\eta_t}\left(1 - \frac{1}{\beta_1}\right) \mathbf{D}_t \odot \mathbf{e}_{t+1}}_{\text{error injection}},
\end{aligned} \tag{13}$$

where $\mathbf{D}_t$ is the diagonal AdamW denominator. ECO uses $\lambda = 1$; we add λ so that $\lambda = 0$ switches the injection off. The tentative point $\tilde{\mathbf{x}}_{t+1}$ exists only within the step: between steps, ECO stores $Q_b(\tilde{\mathbf{x}}_{t+1})$ and the two moments. Here $\mathbf{e}_{t+1}$ is exactly the residual of Section 3, and $\lambda = 0$ with stochastic rounding recovers QAR with the uniform amplifier.

Results. We swept the learning rate for ECO with SR and RTN at each bit width (Figure 15). Table B.3 reports final C4 endpoint loss for seed 0 at the selected settings. ECO has higher final loss than QAT and QAR-γ in our single-seed run, although the difference is small at 8 bits.

Table B.3: Final C4 endpoint loss at 8000 steps of Qwen-2.5-0.5B post-training on fixed weight grids (seed 0), at each method's selected learning rate.

Bits	ECO	rounding, η_0	QAT	QAR-γ
2	6.0077	SR, 1.6×10^{-3}	5.9229	**5.7400**
4	3.6601	RTN, 2×10^{-5}	**3.2267**	3.2813
8	2.9777	RTN, 10^{-5}	2.9745	**2.9736**

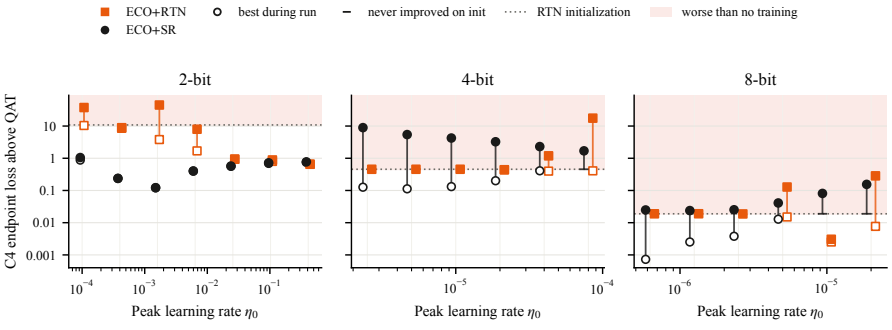


Figure 15: ECO during Qwen-2.5-0.5B post-training with fixed weight grids (seed 0). The vertical axis is C4 endpoint loss above QAT. Filled markers show final loss; hollow markers show the lowest loss observed during each run. The dotted line marks the RTN-initialized endpoint, and the shaded region marks loss above initialization.

Residual alignment. ECO motivates its injection rule with an ideal update that uses both the previous residual $\mathbf{e}_t$ and the current residual $\mathbf{e}_{t+1}$. Its practical rule replaces $\mathbf{e}_t$ with $\mathbf{e}_{t+1}$, motivated by similarity between consecutive residuals (Nikdan et al., 2026). With fresh stochastic rounding

away from saturation on a fixed codebook, the new residual has mean zero conditional on the tentative update and the past. Since $\mathbf{e}_t$ is determined by the past, $\mathbb{E}[\langle \mathbf{e}_t, \mathbf{e}_{t+1} \rangle] = 0$.

Figure 16 measures consecutive-residual alignment and code crossings in our runs. Under SR, alignment is close to 0 at every 4-bit and 8-bit learning rate. At 2 bits it rises toward 1 alongside the saturated fraction, where the unbiased-rounding calculation no longer applies.

Under RTN, residual alignment is close to 1 at every bit width. Such alignment can coexist with a systematic rounding bias. ECO's analysis gives a larger limiting error term in its squared-gradient bound for RTN than for SR when their error scales are comparable; high alignment alone therefore does not guarantee a small optimization error. ECO also reports that SR generally performs better in its experiments (Nikdan et al., 2026).

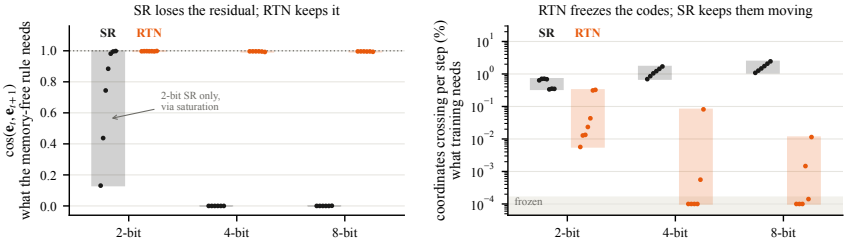


Figure 16: Residual alignment and code motion during Qwen-2.5-0.5B post-training with fixed weight grids (seed 0). Bars span each bit width's learning rate sweep; dots show individual runs. Left: cosine similarity between consecutive residuals. Right: percentage of coordinates crossing per step; the shaded band marks runs with no measurable crossings.

Rare, large corrections under SR. Under SR, a small displacement can still produce a code crossing. For an interior coordinate with tentative displacement $d_i \ll s_i$, the crossing probability is d_i/s_i. On a crossing, the residual has magnitude $s_i - d_i$ and enters the moment with the $1/\eta_t$ factor in (13). For a fixed tentative direction and preconditioner, reducing the learning rate makes crossings rarer while increasing the correction from each crossing. The conditional residual variance is $d_i(s_i - d_i)$, so under these fixed quantities the variance of the injected moment correction scales as $1/\eta_t$ as η_t decreases. Since the corrected moment enters subsequent updates, rare crossings can substantially perturb later update directions. This gives a possible explanation for why several moving SR runs at 4 and 8 bits finish above their best observed endpoint loss (Figure 15).

Frozen codes under RTN. Under RTN, small learning rates freeze: at 4 bits the crossing counter is exactly zero for $\eta_0 \le 10^{-5}$, and those runs end bit-identical to their initialization. ECO rebuilds the tentative point from the stored code each step, so a displacement below half a level leaves the code unchanged. Its effect accumulates in the moment. While no code flips, the residual is exactly the tentative AdamW displacement. Substituting it into (13) gives $\hat{\mathbf{m}}_{t+1} = \tilde{\mathbf{m}}_{t+1}/\beta_1$, hence $\tilde{\mathbf{m}}_{t+2} = \tilde{\mathbf{m}}_{t+1} + (1-\beta_1)\mathbf{g}_{t+1}$. This turns the moving average into a running sum. Starting from zero moments, before the first crossing a code can move only when

$$\eta \left| (\mathbf{D}_t^{-1} \tilde{\mathbf{m}}_t)_i \right| \;=\; \eta(1-\beta_1) \cdot \left| \left(\mathbf{D}_t^{-1} \sum_{u=1}^{t} \mathbf{g}_u \right)_i \right| \;>\; \frac{s_i}{2} \tag{14}$$

for some coordinate i and some $t \le T$, where the second form substitutes the running sum. Since the preconditioner $\mathbf{D}_t$ is the gradient's own root-mean-square scale, each term of the sum contributes magnitude of order one, and the left side grows to about $\eta T(1-\beta_1)$. With $T = 8000$, $\beta_1 = 0.9$ and a median 4-bit half-level $s_i/2 = 5.20 \times 10^{-3}$, no code can move at all unless $\eta > 6.5 \times 10^{-6}$. The first run whose codes move has $\eta_0 = 2 \times 10^{-5}$. The estimate is optimistic for two reasons: η_0 is the peak of a decaying schedule rather than a constant learning rate, and cancellation between gradients can slow the accumulation, so the sum in (14) grows more slowly than T. The threshold explains why the smaller-learning rate runs cannot improve the endpoint loss within the training budget. Larger learning rates let codes move, but some of those runs finish above their starting loss (Figure 15).

Comparison with QAR. ECO removes the master copy only after taking an Adam step, so it must feed the discarded displacement back through the moment, which asks the residual to persist and to

be unbiased at the same time. QAR never forms that displacement: it samples code transitions at the endpoint and applies them directly. A smaller learning rate therefore lowers QAR's crossing rate instead of freezing it.

APPENDIX C RESIDUAL-PHASE DISTRIBUTIONS

C.1 A HIGH-RESOLUTION SOURCE OF UNIFORM PHASES

The main text uses residual phases as a conditional model for endpoint crossings. The following elementary bound gives a sufficient condition under which an unclipped uniform scalar quantizer has nearly uniform residual phases.

Proposition 5. *Let X have density p on $\mathbb{R}$, and let $Q_s(x) = s\left\lfloor \frac{x}{s} + \frac{1}{2} \right\rfloor$ be nearest-neighbor uniform quantization with step size $s > 0$. Define $\theta_s = \frac{X - Q_s(X)}{s}$. If p is absolutely continuous on $\mathbb{R}$, then θ_s has density*

$$\pi_s(\theta) = s \sum_{k \in \mathbb{Z}} p\big(s(k+\theta)\big), \qquad \theta \in [-\tfrac{1}{2}, \tfrac{1}{2}),$$

and

$$\sup_{\theta \in [-1/2, 1/2)} |\pi_s(\theta) - 1| \le s\,\mathrm{TV}(p).$$

Here $\mathrm{TV}(p) = \int_{\mathbb{R}} |p'(x)|\,\mathrm{d}x$ denotes the total variation of p on $\mathbb{R}$. Moreover, the boundary-window mass $B_s(\rho) = \int_{\frac{1}{2}-\rho}^{\frac{1}{2}} \pi_s(\theta)\,\mathrm{d}\theta$ satisfies

$$|B_s(\rho) - \rho| \le \rho\, s\,\mathrm{TV}(p), \qquad 0 \le \rho \le 1.$$

For fixed p, these bounds vanish linearly as $s \to 0$. Thus θ_s converges in total variation to $\mathrm{Unif}\big[-\frac{1}{2}, \frac{1}{2}\big)$, and $B_s(\rho) \to \rho$ uniformly for $0 \le \rho \le 1$.

Proof. The residual-phase map folds each cell $\big[s\big(k - \frac{1}{2}\big), s\big(k + \frac{1}{2}\big)\big)$ onto $\big[-\frac{1}{2}, \frac{1}{2}\big)$. On the cell centered at sk, the inverse branch is $x = s(k + \theta)$, so a change of variables contributes $s\,p(s(k+\theta))$ to the density of θ_s. Summing over $k \in \mathbb{Z}$ gives the displayed density. Now fix θ and write $x_k = s(k+\theta)$. The intervals $I_k = \big[x_k - \frac{s}{2}, x_k + \frac{s}{2}\big)$ partition $\mathbb{R}$, and $x_k \in I_k$. Since $\int p(x)\,\mathrm{d}x = 1$,

$$\pi_s(\theta) - 1 = \sum_{k \in \mathbb{Z}} \int_{I_k} \big(p(x_k) - p(x)\big)\,\mathrm{d}x.$$

Because p is absolutely continuous, $p(x_k) - p(x) = \int_x^{x_k} p'(u)\,\mathrm{d}u$ for every $x \in I_k$, so

$$|p(x_k) - p(x)| \le \int_{I_k} |p'(u)|\,\mathrm{d}u, \qquad x \in I_k.$$

Integrating over $x \in I_k$ costs a factor $|I_k| = s$, and summing over k recovers the full integral of $|p'|$ exactly once:

$$|\pi_s(\theta) - 1| \le \sum_{k \in \mathbb{Z}} s \int_{I_k} |p'(u)|\,\mathrm{d}u = s \int_{\mathbb{R}} |p'(u)|\,\mathrm{d}u = s\,\mathrm{TV}(p).$$

The boundary-window bounds follow by integrating this pointwise density bound over intervals of length ρ. □

C.2 FOLDED RESIDUAL PHASES OF PRETRAINED WEIGHTS

In this subsection, we give additional evidence for the pretrained normal-weight phenomenon for Qwen-2.5-0.5B. We collect linear weight tensors, exclude embeddings and the LM head, rescale each layer, and draw weighted samples. Figure 17 reports the pooled diagnostic. After local rescaling, the central pretrained weight mass is well approximated by a centered normal law, which induces the folded residual phase model for finite grids.

The folded-normal phase model. We recall the model of Section 3 with its normalization. Let $X \sim \mathcal{N}(0, \tau^2)$, and let the b-bit grid have spacing $s_b = c/K_b$, where $c > 0$ is the clipping threshold, so that $\lambda_b = \tau/s_b$ is the weight scale in grid-step units. For fixed τ and c, $\lambda_b = K_b\lambda_2$. The pooled interior residual phase has density

$$\pi(\theta) = \frac{\sum_{k=-K_b+1}^{K_b-1} \exp\left(-\frac{(k+\theta)^2}{2\lambda_b^2}\right)}{\int_{-1/2}^{1/2} \sum_{k=-K_b+1}^{K_b-1} \exp\left(-\frac{(k+v)^2}{2\lambda_b^2}\right) \mathrm{d}v}, \qquad \theta \in \left[-\frac{1}{2}, \frac{1}{2}\right).$$

At 2 bits, $K_2 = 1$, so the sum contains only the zero cell and $\pi(\cdot)$ is the density of a truncated normal on $\left[-\frac{1}{2}, \frac{1}{2}\right)$.

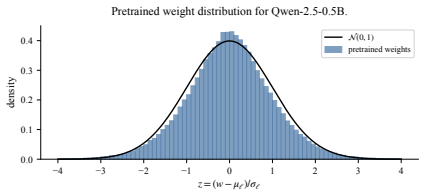


Figure 17: Pooled pretrained weight distribution for Qwen-2.5-0.5B.

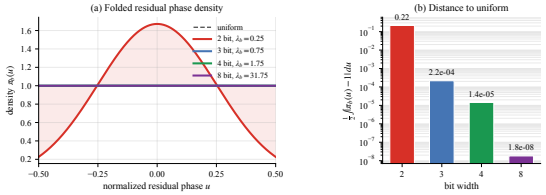


Figure 18: Predicted folded residual phase density $\pi_b(\theta)$ compared with the uniform density.

Figure 18 makes the scale separation explicit. With the 2-bit fit $\lambda_2 = 0.25$, $\pi(\cdot)$ is center-concentrated and has substantial distance from the uniform cell-phase model. The same absolute weight scale makes the 3-, 4-, and 8-bit folded densities nearly uniform, with total-variation distance dropping by roughly three orders of magnitude already at 3 bits. Figure 2 reports the corresponding empirical Qwen-2.5-0.5B residual histograms at the initial checkpoint.

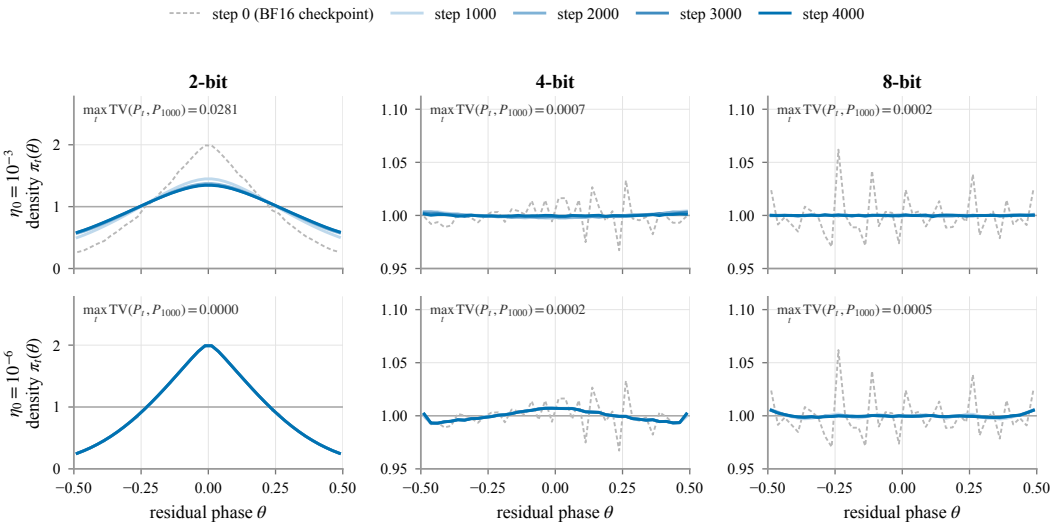


Figure 19: Residual-phase density $\pi_t(\theta)$ over QAT training. Blue curves are steps 1000 to 4000, grey is the step-0 BF16 checkpoint, and each panel reports $\max_t \mathrm{TV}(P_t, P_{1000})$.

Figure 19 compares the full residual-phase density over QAT training. For all four learning rates $\eta_0 \in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$ (the two extremes are shown), the density changes little after the initial adjustment, especially at 4 and 8 bits. As in Appendix B.3, the step-0 curve is coarse because the BF16 lattice is sparser than the phase bins; one optimizer step moves the weights off the lattice, so we measure drift from step 1000.

APPENDIX D PROOFS FOR CROSSING AND ENDPOINT MOTION

D.1 PROOF OF LEMMA 1

Proof of Lemma 1. Fix coordinate i. If $u_i > 0$, then $x_i^+ = x_i - \eta u_i < x_i$. For the fixed coordinate-wise uniform quantizer, decreasing x_i cannot increase the endpoint, so

$$q_i^+ = Q_b(\mathbf{x}^+)_i \le Q_b(\mathbf{x})_i = q_i.$$

Hence $q_i^+ - q_i \le 0$, and multiplying by $u_i > 0$ gives $u_i(q_i^+ - q_i) \le 0$. If $u_i < 0$, then $x_i^+ > x_i$, so the same fixed grid ordering gives $q_i^+ \ge q_i$, and again $u_i(q_i^+ - q_i) \le 0$. If $u_i = 0$, the product is zero. If $g_i u_i \ge 0$ coordinatewise, then g_i has the same sign as u_i whenever both are nonzero, so $g_i(q_i^+ - q_i) \le 0$ as well. Summing over coordinates proves the inner-product statement. □

D.2 PROOF OF PROPOSITION 2

Proof of Proposition 2. Under the stated small-step condition, a crossed active coordinate moves by one neighboring grid level in the direction opposite to u_i, while a non-crossed coordinate does not move. Hence

$$q_i^+ - q_i = -s_i \operatorname{sign}(u_i) J_i.$$

Applying Assumption 2 with $\mathbf{x} = \mathbf{q}$ and $\mathbf{y} = \mathbf{q}^+$ gives

$$f(\mathbf{q}^+) - f(\mathbf{q}) \le \langle \mathbf{g}, \mathbf{q}^+ - \mathbf{q} \rangle + \frac{1}{2} \sum_i L_i (q_i^+ - q_i)^2.$$

Substituting the realized jump representation yields

$$\langle \mathbf{g}, \mathbf{q}^+ - \mathbf{q} \rangle = -\sum_i J_i s_i \operatorname{sign}(u_i) g_i, \qquad (q_i^+ - q_i)^2 = J_i s_i^2,$$

which proves (3). □

D.3 CROSSING PROBABILITY OF A SINGLE COORDINATE

Lemma 6 (One-coordinate crossing probability)**.** *Consider one non-saturated coordinate i with*

$$r_i \in \left[-\frac{s_i}{2}, \frac{s_i}{2}\right), \qquad x_i^+ = x_i - \eta u_i, \qquad \eta |u_i| < s_i.$$

The endpoint crosses only when r_i lies in the boundary window

$$I_i(u_i) = \begin{cases} \left[-\frac{s_i}{2}, -\frac{s_i}{2} + \eta u_i\right), & u_i > 0, \\ \left[\frac{s_i}{2} + \eta u_i, \frac{s_i}{2}\right), & u_i < 0, \\ \varnothing, & u_i = 0. \end{cases}$$

On this event,

$$q_i^+ - q_i = -s_i \operatorname{sign}(u_i).$$

Hence, if $\pi_i(\cdot \mid \mathbf{q})$ is the conditional density of r_i, then the crossing probability is

$$p_i = \int_{I_i(u_i)} \pi_i(v \mid \mathbf{q}) \, \mathrm{d}v, \qquad \mathbb{E}[q_i^+ - q_i \mid \mathbf{q}] = -s_i \operatorname{sign}(u_i) \, p_i. \tag{15}$$

Proof of Lemma 6. Fix coordinate i. If $u_i > 0$, the update moves left. A left crossing occurs precisely when

$$r_i - \eta u_i < -\frac{s_i}{2},$$

which is equivalent to the stated interval. Since one threshold is crossed, the endpoint moves by $q_i^+ - q_i = -s_i$. The case $u_i < 0$ is symmetric: the update moves right, and the crossing condition is

$$r_i - \eta u_i \ge \frac{s_i}{2}.$$

The expectation follows by multiplying the signed jump by its crossing probability. Integrating the conditional density $\pi_i(\cdot \mid \mathbf{q})$ over the crossing interval gives (15). □

D.4 Proof of Theorem 3

Proof of Theorem 3. Under the stated stepsize condition, the argument in the proof of Proposition 2 gives $q_i^+ - q_i = -s_i \operatorname{sign}(u_i) J_i$, which is (4), together with the realized bound (3). Taking conditional expectations and using $\mathbb{E}[J_i \mid \eta, \mathbf{q}, \mathbf{u}] = p_i(\eta, \mathbf{q}, \mathbf{u})$ from Model 1 proves (5). Under Model 2, $p_i(\eta, \mathbf{q}, \mathbf{u}) = \eta |u_i| / s_i$, which gives (6). □

Appendix E Power-Amplifier Guarantees

E.1 Proof of Theorem 4

Recall $a_i = M_i |u_i|$ and the power score (8) from Section 4.2, where $M_i = \mathbf{1}\{k_i - \operatorname{sign}(u_i) \in \mathcal{K}_i\}$ at an endpoint $q_i = s_G k_i$. By (9), the uncapped transition probability is

$$p_i = \frac{\eta a_i b_{\gamma,i}}{s_G}. \tag{16}$$

Lemma 7 (Power-score properties). *For every group G and $\gamma \geq 0$,*

$$\sum_{i \in G} b_{\gamma,i} = |G|, \qquad \sum_{i \in G} b_{\gamma,i} a_i \geq \sum_{i \in G} a_i, \qquad \sum_{i \in G} b_{\gamma,i} a_i^2 \geq \sum_{i \in G} a_i^2. \tag{17}$$

For $\gamma > 0$, both inequalities are strict exactly when the signals in the group are not all equal.

Proof. The claim is immediate for $\gamma = 0$ and for an all-zero group. Otherwise, the normalization in (8) gives $\sum_i b_{\gamma,i} = |G|$. For $r \in \{1, 2\}$, the maps $z \mapsto z^\gamma$ and $z \mapsto z^r$ are nondecreasing on $\mathbb{R}_+$, so the finite-sum Chebyshev inequality gives

$$\frac{1}{|G|} \sum_{i \in G} a_i^{\gamma + r} \geq \left(\frac{1}{|G|} \sum_{i \in G} a_i^\gamma \right) \left(\frac{1}{|G|} \sum_{i \in G} a_i^r \right).$$

Dividing by the first parenthesized factor proves both inequalities in (17). Strictness follows when two signals differ. □

Proposition 8 (One-step GD bound). *Suppose the route is the exact gradient and the probabilities are uncapped. For $\mathcal{A}_\gamma$ with any $\gamma \geq 0$, let $b_i = b_{\gamma,i}$. Then*

$$\begin{aligned} \mathbb{E}[f(\mathbf{q}^+) - f(\mathbf{q}) \mid \mathbf{q}] &\leq -\eta \sum_G \sum_{i \in G} b_i a_i^2 + \frac{\eta}{2} \sum_G L_G s_G \sum_{i \in G} b_i a_i \\ &\leq -\frac{\eta}{2} \|\mathbf{g}_{\mathcal{Q}}(\mathbf{q})\|_2^2 + \frac{\eta}{8} \sum_G |G| L_G^2 s_G^2. \end{aligned} \tag{18}$$

Proof. Since the power score redistributes transition probability within each group, we bound the curvature groupwise: with $L_G = \max_{i \in G} L_i$, Assumption 2 implies

$$f(\mathbf{y}) \leq f(\mathbf{q}) + \langle \nabla f(\mathbf{q}), \mathbf{y} - \mathbf{q} \rangle + \frac{1}{2} \sum_G L_G \|\mathbf{y}_G - \mathbf{q}_G\|_2^2. \tag{19}$$

For the exact-gradient route, $a_i = |(\mathbf{g}_{\mathcal{Q}}(\mathbf{q}))_i|$. An accepted jump in coordinate $i \in G$ contributes $-s_G a_i$ to the linear term of (19) and s_G^2 to its squared displacement. Substituting $p_i = \eta a_i b_i / s_G$ gives the first line. For $c_G = L_G s_G$, the scalar inequality

$$-a_i^2 + \frac{c_G}{2} a_i \leq -\frac{1}{2} a_i^2 + \frac{1}{8} c_G^2$$

and $\sum_{i \in G} b_i = |G|$ give the second line. The final line follows from Lemma 7, with equality at $\gamma = 0$. □

Proposition 9 (One-step signGD bound). *Suppose $\mathbf{u} = \operatorname{sign}(\nabla f(\mathbf{q}))$ and the probabilities are uncapped. For $\mathcal{A}_\gamma$ with any $\gamma \geq 0$, let $b_i = b_{\gamma,i}$. Then*

$$\begin{aligned} \mathbb{E}[f(\mathbf{q}^+) - f(\mathbf{q}) \mid \mathbf{q}] &\leq -\eta \sum_G \sum_{i \in G} b_i \left| \left(\mathbf{g}_{\mathcal{Q}}(\mathbf{q})\right)_i \right| + \frac{\eta}{2} \sum_G L_G s_G \sum_{i \in G} b_i a_i \\ &\leq -\eta \|\mathbf{g}_{\mathcal{Q}}(\mathbf{q})\|_1 + \frac{\eta}{2} \sum_G |G| L_G s_G. \end{aligned} \tag{20}$$

Proof. Here $a_i = M_i|u_i| \in \{0,1\}$, and $a_i = 1$ exactly when the corresponding component of $\mathbf{g}_{\mathcal{Q}}(\mathbf{q})$ is nonzero. Substituting $p_i = \eta a_i b_i / s_G$ into (19) gives the first line. For $\gamma = 0$, active coordinates have $b_i = 1$ and $\sum_{i\in G} b_i a_i \le |G|$. For $\gamma > 0$, if $m_G = \sum_{i\in G} a_i > 0$, every active coordinate has $b_i = |G|/m_G \ge 1$ and $\sum_i b_i a_i = |G|$; when $m_G = 0$, the group does not move. Thus the weighted first-order signal is at least the unweighted feasible-gradient ℓ_1 norm, while the curvature mass is at most $|G|$ in every group. □

Proof of Theorem 4. Since $a_{t,i} = M_{t,i}|u_{t,i}| \le U$, the condition $\eta U b_{t,i} \le s_G$ implies $\eta a_{t,i} b_{t,i}/s_G \le 1$. Hence the coordinatewise minimum in Algorithm 1 is inactive. Therefore Proposition 8 applies to the GD route and Proposition 9 applies to the signGD route at every step. For GD, keep the weighted second line of (18), take total expectation, and sum over $t = 0, \dots, T-1$. Since $f(\mathbf{q}_T) \ge f^*_{\mathcal{Q}}$, this gives

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\sum_i b_{t,i}\big(\mathbf{g}_{\mathcal{Q}}(\mathbf{q}_t)\big)_i^2\right] \le \frac{2(f(\mathbf{q}_0) - f^*_{\mathcal{Q}})}{\eta T} + \frac{1}{4}\sum_G |G| L_G^2 s_G^2.$$

This proves the second inequality in (11); the first follows by applying Lemma 7 within each group before taking expectations. For signGD, the same summation applied to (20) proves (12). □

E.2 Tightness of the Finite-Grid Floors

For arbitrary dimension d and positive L_i, s_i, consider

$$\mathcal{Q} = \prod_{i=1}^{d}\{-s_i, 0, s_i\}, \qquad f(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{d} L_i\left(x_i - \frac{s_i}{2}\right)^2.$$

Its Hessian is $\mathrm{diag}(L_1, \dots, L_d)$, so the coordinate-smoothness constants are exactly L_i. Every $\mathbf{q} \in \prod_i\{0, s_i\}$ is a global minimizer over $\mathcal{Q}$. Moreover, the descent-sign move toggles coordinate i between 0 and s_i, and hence

$$\mathbf{g}_{\mathcal{Q}}(\mathbf{q}) = \nabla f(\mathbf{q}), \qquad \|\mathbf{g}_{\mathcal{Q}}(\mathbf{q})\|_2^2 = \frac{1}{4}\|\mathbf{L}\odot\mathbf{s}\|_2^2, \qquad \|\mathbf{g}_{\mathcal{Q}}(\mathbf{q})\|_1 = \frac{1}{2}\|\mathbf{L}\odot\mathbf{s}\|_1.$$

With exact gradients and $0 < \eta \le 2/\max_i L_i$, the QAR-0 SGD route moves coordinate i with probability $\eta L_i/2$, so a trajectory initialized in $\prod_i\{0, s_i\}$ remains there. The initial finite-grid optimality gap and noise terms are zero, while the averaged feasible-gradient measure equals the SGD floor for every T. The same trajectory under the QAR-0 signSGD route, with $0 < \eta \le \min_i s_i$, attains its ℓ_1 floor for every T.

Appendix F Stochastic QAR Guarantees

To analyze QAR under stochastic gradient noise, we follow Bernstein et al. (2018) and adopt the standard assumption that the mini-batch gradient $\widehat{\mathbf{g}}(\mathbf{q})$ is unbiased with bounded coordinatewise variance.

Assumption 3 (Stochastic gradient noise)**.** *At each endpoint $\mathbf{q}$, the mini-batch gradient is unbiased and has coordinate variance*

$$\mathbb{E}[(\widehat{g}_i(\mathbf{q}) - g_i(\mathbf{q}))^2 \mid \mathbf{q}] \le \sigma_i^2/n.$$

Given $(\mathbf{q}_t, \widehat{\mathbf{g}}_t)$, Algorithm 1 samples the jump variables coordinatewise with probabilities $\mathbf{p}_t$, using fresh routing randomness.

F.1 Baseline Stochastic QAR Bounds

Theorem 10 (Stochastic QAR finite-grid bounds)**.** *Suppose Assumptions 2 and 3 hold, and write $f^*_{\mathcal{Q}} = \min_{\mathbf{q}\in\mathcal{Q}} f(\mathbf{q})$. Fix $\eta > 0$ and an integer $T \ge 1$. For each scale-sharing group G, let $s_i = s_G$ for $i \in G$, $L_G = \max_{i\in G} L_i$, and $\boldsymbol{\sigma}_G = (\sigma_i)_{i\in G}$. For the SGD route $\mathbf{u}_t = \widehat{\mathbf{g}}(\mathbf{q}_t)$, use $\mathcal{A}_\gamma$ with $\gamma = 0$ and suppose $\eta|\widehat{g}_{t,i}| \le s_i$ almost surely. Then*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\mathbf{g}_{\mathcal{Q}}(\mathbf{q}_t)\|_2^2 \le \frac{2(f(\mathbf{q}_0) - f^*_{\mathcal{Q}})}{\eta T} + \frac{1}{4}\|\mathbf{L}\odot\mathbf{s}\|_2^2 + \frac{\|\boldsymbol{\sigma}\|_2^2}{n}. \tag{21}$$

For the signSGD route $\mathbf{u}_t = \operatorname{sign}(\widehat{\mathbf{g}}(\mathbf{q}_t))$*, use* $\mathcal{A}_\gamma$ *for any* $\gamma \geq 0$*, with* $\eta \leq \min_G s_G$ *when* $\gamma = 0$ *and* $\eta \leq \min_G(s_G/|G|)$ *when* $\gamma > 0$*. Then*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\mathbf{g}_{\mathcal{Q}}(\mathbf{q}_t)\|_1 \leq \frac{f(\mathbf{q}_0) - f^*_{\mathcal{Q}}}{\eta T} + \begin{cases} \dfrac{1}{2}\|\mathbf{L}\odot\mathbf{s}\|_1 + \dfrac{\|\boldsymbol{\sigma}\|_1}{\sqrt{n}}, & \gamma = 0, \\ \dfrac{1}{2}\displaystyle\sum_G |G| L_G s_G + \frac{1}{\sqrt{n}}\sum_G |G|\|\boldsymbol{\sigma}_G\|_1, & \gamma > 0, \end{cases} \tag{22}$$

Proof. Fix $\mathbf{q}_t$, write $\mathbf{g}_t = \nabla f(\mathbf{q}_t)$, $\widehat{\mathbf{g}}_t = \mathbf{g}_t + \boldsymbol{\xi}_t$, and $\mathbf{h}_t = \mathbf{g}_{\mathcal{Q}}(\mathbf{q}_t)$. Also write $\mathbf{z}_t = \operatorname{sign}(\widehat{\mathbf{g}}_t)$. The coordinate mask $M_{t,i}$ keeps only feasible one-cell moves, so the endpoint always remains in $\mathcal{Q}$.

SGD route. For $\mathbf{u}_t = \widehat{\mathbf{g}}_t$, conditioning on $(\mathbf{q}_t, \widehat{\mathbf{g}}_t)$ and applying Assumption 2 gives

$$\mathbb{E}_{\mathbf{J}}[f(\mathbf{q}_{t+1}) - f(\mathbf{q}_t) \mid \mathbf{q}_t, \widehat{\mathbf{g}}_t] \leq -\eta\sum_i g_{t,i}\widehat{g}_{t,i}M_{t,i} + \frac{\eta}{2}\sum_i L_i s_i |\widehat{g}_{t,i}| M_{t,i}.$$

We bound the drift and curvature terms together. If $h_{t,i} = g_{t,i}$, the true descent direction is feasible, and the mask either keeps the sample or removes one with $g_{t,i}\widehat{g}_{t,i} \leq 0$. Hence, by unbiasedness and Cauchy–Schwarz,

$$\mathbb{E}[g_{t,i}\widehat{g}_{t,i}M_{t,i} \mid \mathbf{q}_t] \geq h_{t,i}^2,$$
$$\mathbb{E}[M_{t,i}|\widehat{g}_{t,i}| \mid \mathbf{q}_t] \leq \sqrt{h_{t,i}^2 + \frac{\sigma_i^2}{n}}.$$

Combining these inequalities and completing the square gives

$$\begin{aligned} \mathbb{E}\left[-g_{t,i}M_{t,i}\widehat{g}_{t,i} + \frac{L_i s_i}{2}M_{t,i}|\widehat{g}_{t,i}| \,\middle|\, \mathbf{q}_t\right] &\leq -h_{t,i}^2 + \frac{L_i s_i}{2}\sqrt{h_{t,i}^2 + \frac{\sigma_i^2}{n}} \\ &\leq -\frac{1}{2}h_{t,i}^2 + \frac{1}{8}L_i^2 s_i^2 + \frac{1}{2}\frac{\sigma_i^2}{n}. \end{aligned} \tag{23}$$

Otherwise $h_{t,i} = 0 \neq g_{t,i}$ and the true descent direction is blocked at a boundary. By symmetry take the upper boundary, so $g_{t,i} < 0$ and the mask keeps exactly the feasible samples, $M_{t,i}\widehat{g}_{t,i} = (\widehat{g}_{t,i})_+$. Since $\xi_{t,i} + \frac{L_i s_i}{2} = \widehat{g}_{t,i} + \left(|g_{t,i}| + \frac{L_i s_i}{2}\right)$ and $4uv \leq (u+v)^2$,

$$-g_{t,i}M_{t,i}\widehat{g}_{t,i} + \frac{L_i s_i}{2}M_{t,i}|\widehat{g}_{t,i}| = \left(|g_{t,i}| + \frac{L_i s_i}{2}\right)(\widehat{g}_{t,i})_+ \leq \frac{1}{4}\left(\xi_{t,i} + \frac{L_i s_i}{2}\right)^2.$$

Taking conditional expectation and using unbiasedness gives

$$\mathbb{E}\left[-g_{t,i}M_{t,i}\widehat{g}_{t,i} + \frac{L_i s_i}{2}M_{t,i}|\widehat{g}_{t,i}| \,\middle|\, \mathbf{q}_t\right] \leq \frac{\sigma_i^2}{4n} + \frac{L_i^2 s_i^2}{16} \leq \frac{1}{2}\frac{\sigma_i^2}{n} + \frac{1}{8}L_i^2 s_i^2,$$

which is the final bound in (23) with $h_{t,i} = 0$. Hence (23) holds for every coordinate, and summing these coordinate bounds gives

$$\mathbb{E}[f(\mathbf{q}_{t+1}) - f(\mathbf{q}_t)] \leq -\frac{\eta}{2}\mathbb{E}\|\mathbf{h}_t\|_2^2 + \frac{\eta}{8}\|\mathbf{L}\odot\mathbf{s}\|_2^2 + \frac{\eta\|\boldsymbol{\sigma}\|_2^2}{2n}.$$

Summing over $t = 0, \ldots, T-1$, using $f(\mathbf{q}_T) \geq f^*_{\mathcal{Q}}$, and dividing by $\eta T/2$ proves (21).

signSGD route. Recall that $\mathbf{z}_t = \operatorname{sign}(\widehat{\mathbf{g}}_t)$. For $\gamma = 0$, every active score is $b_{t,i} = 1$. For $\gamma > 0$, each active score in a group G equals $|G|$ divided by the number of active coordinates in that group. Thus $1 \leq b_{t,i} \leq |G|$ on active coordinates, and in either case

$$\sum_{i\in G} b_{t,i}M_{t,i}|z_{t,i}| \leq |G|.$$

The stated stepsize conditions therefore keep all probabilities uncapped. Conditioning on $(\mathbf{q}_t, \widehat{\mathbf{g}}_t)$ and applying Assumption 2 gives

$$\mathbb{E}_{\mathbf{J}}[f(\mathbf{q}_{t+1}) - f(\mathbf{q}_t) \mid \mathbf{q}_t, \widehat{\mathbf{g}}_t] \leq -\eta\sum_i b_{t,i}g_{t,i}z_{t,i}M_{t,i} + \frac{\eta}{2}\sum_G s_G\sum_{i\in G} L_i b_{t,i}M_{t,i}|z_{t,i}|.$$

For every $i \in G$, a case split over a correct sign, an incorrect sign, and $z_{t,i} = 0$ gives

$$b_{t,i} g_{t,i} z_{t,i} M_{t,i} \geq |h_{t,i}| - \begin{cases} 2|g_{t,i}| \mathbf{1}\{z_{t,i} \neq \operatorname{sign}(g_{t,i})\}, & \gamma = 0, \\ 2|G||g_{t,i}| \mathbf{1}\{z_{t,i} \neq \operatorname{sign}(g_{t,i})\}, & \gamma > 0. \end{cases}$$

This also covers a boundary where the true descent direction is infeasible: then $h_{t,i} = 0$, and a harmful inward move requires a wrong sign. For $g_{t,i} \neq 0$, Cantelli's one-sided inequality gives

$$\Pr(z_{t,i} \neq \operatorname{sign}(g_{t,i}) \mid \mathbf{q}_t) \leq \frac{\sigma_i^2/n}{g_{t,i}^2 + \sigma_i^2/n},$$

$$2|g_{t,i}| \Pr(z_{t,i} \neq \operatorname{sign}(g_{t,i}) \mid \mathbf{q}_t) \leq \frac{2|g_{t,i}|\sigma_i^2/n}{g_{t,i}^2 + \sigma_i^2/n} \leq \frac{\sigma_i}{\sqrt{n}}.$$

The event includes both a wrong sign and a zero sign, and the last inequality uses $2ab \leq a^2 + b^2$. Coordinates with $g_{t,i} = 0$ contribute zero, so

$$\mathbb{E}\left[\sum_i b_{t,i} g_{t,i} z_{t,i} M_{t,i} \,\middle|\, \mathbf{q}_t \right] \geq \|\mathbf{g}_{\mathcal{Q}}(\mathbf{q}_t)\|_1 - \begin{cases} \dfrac{\|\boldsymbol{\sigma}\|_1}{\sqrt{n}}, & \gamma = 0, \\ \dfrac{1}{\sqrt{n}} \displaystyle\sum_G |G| \|\boldsymbol{\sigma}_G\|_1, & \gamma > 0. \end{cases}$$

For the curvature term, $\gamma = 0$ retains the sharper bound $\sum_i L_i s_i$, while $\gamma > 0$ is bounded by $\sum_G |G| L_G s_G$. Taking expectation over the stochastic gradient therefore gives

$$\mathbb{E}[f(\mathbf{q}_{t+1}) - f(\mathbf{q}_t) \mid \mathbf{q}_t] \leq -\eta \|\mathbf{h}_t\|_1 + \begin{cases} \dfrac{\eta}{2} \|\mathbf{L} \odot \mathbf{s}\|_1 + \dfrac{\eta \|\boldsymbol{\sigma}\|_1}{\sqrt{n}}, & \gamma = 0, \\ \dfrac{\eta}{2} \displaystyle\sum_G |G| L_G s_G + \dfrac{\eta}{\sqrt{n}} \sum_G |G| \|\boldsymbol{\sigma}_G\|_1, & \gamma > 0. \end{cases}$$

Summing over $t = 0, \ldots, T-1$, using $f(\mathbf{q}_T) \geq f_{\mathcal{Q}}^*$, and dividing by ηT proves (22). □

F.2 Noisy Power-Routing Guarantee

Consider the stochastic full-magnitude route $\mathbf{u} = \widehat{\mathbf{g}}(\mathbf{q})$. Write $\widehat{\mathbf{g}} = \mathbf{g} + \boldsymbol{\xi}$, and construct the feasibility mask and power score from the same sampled gradient. For $i \in G$, let

$$\widehat{a}_i = M_i |\widehat{g}_i|, \qquad b_i = b_{\gamma,i}(\widehat{\mathbf{a}}).$$

The effect of amplification can be summarized by a net margin, defined by

$$\Delta_{\gamma,G}(\mathbf{q}) := \mathbb{E}\left[\sum_{i \in G} (b_i - 1) \left(\widehat{a}_i^2 - \frac{1}{2} L_i s_i \widehat{a}_i \right) \,\middle|\, \mathbf{q} \right] - \frac{\|\boldsymbol{\sigma}_G\|_2}{\sqrt{n}} \left(\mathbb{E}\left[\sum_{i \in G} (b_i - 1)^2 \widehat{a}_i^2 \,\middle|\, \mathbf{q} \right] \right)^{1/2}. \tag{24}$$

Within each group, the first term is the additional first-order signal captured by power routing minus its curvature cost; the second accounts for selecting the routing scores from noisy gradients. Let $\Delta_\gamma(\mathbf{q}) = \sum_G \Delta_{\gamma,G}(\mathbf{q})$.

Theorem 11 (Noisy QAR-γ trade-off)**.** *Suppose Assumptions 2 and 3 hold. Run Algorithm 1 with $\mathbf{u}_t = \widehat{\mathbf{g}}(\mathbf{q}_t)$ and $\mathcal{A}_\gamma$, $\gamma \geq 0$, and assume $\eta \widehat{a}_{t,i} b_{t,i} \leq s_G$ almost surely for every $t < T$, every group G, and every $i \in G$. Let $f_{\mathcal{Q}}^* = \min_{\mathbf{q} \in \mathcal{Q}} f(\mathbf{q})$. Then*

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\|\mathbf{g}_{\mathcal{Q}}(\mathbf{q}_t)\|_2^2 \leq \frac{2(f(\mathbf{q}_0) - f_{\mathcal{Q}}^*)}{\eta T} + \frac{1}{4} \|\mathbf{L} \odot \mathbf{s}\|_2^2 + \frac{\|\boldsymbol{\sigma}\|_2^2}{n} - \frac{2}{T} \sum_{t=0}^{T-1} \mathbb{E}\Delta_\gamma(\mathbf{q}_t). \tag{25}$$

At $\gamma = 0$, $b_i = 1$, so $\Delta_0 = 0$ and the first three terms on the right-hand side recover the QAR-0 certificate in Theorem 10. Power amplification gives a strictly sharper certificate whenever $\sum_{t=0}^{T-1} \mathbb{E}\Delta_\gamma(\mathbf{q}_t) > 0$. The uncapped assumption is needed because otherwise the cap can truncate the largest power scores, and the deterministic comparison in Lemma 7 need not be preserved.

Proof. Condition on $(\mathbf{q}, \widehat{\mathbf{g}})$, and let $v_i = M_i g_i \operatorname{sign}(\widehat{g}_i)$. Let $\mathcal{U}(\mathbf{q}, \widehat{\mathbf{g}})$ denote the one-step upper bound obtained with $b_i = 1$. Plugging the probability $p_i = \eta \widehat{a}_i b_i / s_G$ into the expectation, we have

$$\mathbb{E}_{\mathbf{J}}[f(\mathbf{q}^+) - f(\mathbf{q}) \mid \mathbf{q}, \widehat{\mathbf{g}}] \le \mathcal{U}(\mathbf{q}, \widehat{\mathbf{g}}) - \eta \sum_G \sum_{i \in G} (b_i - 1) \left(\widehat{a}_i^2 - \frac{1}{2} L_i s_i \widehat{a}_i \right) + \eta \sum_G \sum_{i \in G} |b_i - 1| \widehat{a}_i |\xi_i|. \tag{26}$$

Indeed,

$$|v_i - \widehat{a}_i| = M_i |\operatorname{sign}(\widehat{g}_i)| \, |g_i - \widehat{g}_i| \le |\xi_i|.$$

Conditional Cauchy–Schwarz and Assumption 3 imply

$$\mathbb{E}\left[\sum_G \sum_{i \in G} |b_i - 1| \widehat{a}_i |\xi_i| \,\middle|\, \mathbf{q} \right] \le \frac{1}{\sqrt{n}} \sum_G \left(\mathbb{E}\left[\sum_{i \in G} (b_i - 1)^2 \widehat{a}_i^2 \,\middle|\, \mathbf{q} \right] \right)^{1/2} \|\boldsymbol{\sigma}_G\|_2.$$

Applying the coordinate estimate established in the SGD part of Theorem 10 to $\mathcal{U}$ yields

$$\mathbb{E}[\mathcal{U}(\mathbf{q}, \widehat{\mathbf{g}}) \mid \mathbf{q}] \le -\frac{\eta}{2} \|\mathbf{g}_{\mathcal{Q}}(\mathbf{q})\|_2^2 + \frac{\eta}{8} \|\mathbf{L} \odot \mathbf{s}\|_2^2 + \frac{\eta \|\boldsymbol{\sigma}\|_2^2}{2n}.$$

Taking conditional expectation in (26) and using (24) therefore gives

$$\mathbb{E}[f(\mathbf{q}^+) - f(\mathbf{q}) \mid \mathbf{q}] \le -\frac{\eta}{2} \|\mathbf{g}_{\mathcal{Q}}(\mathbf{q})\|_2^2 + \frac{\eta}{8} \|\mathbf{L} \odot \mathbf{s}\|_2^2 + \frac{\eta \|\boldsymbol{\sigma}\|_2^2}{2n} - \eta \Delta_\gamma(\mathbf{q}).$$

Summing over $t < T$, using $f(\mathbf{q}_T) \ge f^*_{\mathcal{Q}}$, and dividing by $\eta T/2$ proves (25). □

F.3 When Does Amplification Pay Off?

The selection-noise penalty in (24) is nonnegative and carries an explicit $n^{-1/2}$ factor, so for large batches the first term determines whether amplification pays off. To make this condition explicit, fix a realized nonconstant group of magnitudes a_i, write $m = |G|$, and suppose $L_i = L_G$ within the group. Define

$$R_\gamma = \frac{\sum_{i \in G} (b_{\gamma,i} - 1) a_i^2}{\sum_{i \in G} (b_{\gamma,i} - 1) a_i}, \tag{27}$$

the signal gained per unit of additional crossing mass. Its denominator is positive by Lemma 12 below, and the group contribution to the first term of Δ_γ factors as

$$\sum_{i \in G} (b_{\gamma,i} - 1) \left(a_i^2 - \frac{1}{2} L_G s_G a_i \right) = \left[\sum_{i \in G} (b_{\gamma,i} - 1) a_i \right] \left(R_\gamma - \frac{L_G s_G}{2} \right).$$

Thus amplification pays its curvature cost exactly when

$$R_\gamma > \frac{L_G s_G}{2}.$$

The quantity R_γ measures the within-group signal imbalance that the amplifier can exploit. Lemma 12 shows how this quantity changes with the power exponent.

Lemma 12 (Monotonicity of the imbalance ratio)**.** *Fix a nonconstant group of nonnegative magnitudes a_i. For $\gamma > 0$, the denominator of R_γ in (27) is positive and R_γ is nondecreasing in γ. It is strictly increasing when the group has at least three distinct magnitudes and constant when it has exactly two.*

Proof. It suffices to treat positive a_i; zero entries follow by continuity. Write $m = |G|$ and $S_r = \sum_{i \in G} a_i^r$. Substituting $b_{\gamma,i} = m a_i^\gamma / S_\gamma$ gives

$$\sum_i (b_{\gamma,i} - 1) a_i^2 = \frac{m S_{\gamma+2} - S_2 S_\gamma}{S_\gamma}, \qquad \sum_i (b_{\gamma,i} - 1) a_i = \frac{m S_{\gamma+1} - S_1 S_\gamma}{S_\gamma},$$

so that

$$R_\gamma = \frac{mS_{\gamma+2} - S_2 S_\gamma}{mS_{\gamma+1} - S_1 S_\gamma}.$$

The denominator is positive by the finite-sum Chebyshev inequality applied to the similarly ordered sequences (a_i) and (a_i^γ), strictly so because the group is nonconstant. For $0 < \gamma_1 < \gamma_2$, the moment kernel S_{x+y} is totally positive. Indeed, Cauchy–Binet applied to $S_{x+y} = \sum_i e^{x \log a_i} e^{y \log a_i}$ expresses each minor as a sum of products of equally signed exponential Vandermonde determinants. Hence

$$\det \begin{pmatrix} S_0 & S_1 & S_2 \\ S_{\gamma_1} & S_{\gamma_1+1} & S_{\gamma_1+2} \\ S_{\gamma_2} & S_{\gamma_2+1} & S_{\gamma_2+2} \end{pmatrix} \geq 0.$$

Subtracting S_{γ_j}/m times the first row from row $j+1$ shows that this determinant is nonnegative exactly when $R_{\gamma_2} \geq R_{\gamma_1}$. The determinant is strict with at least three distinct magnitudes; with two distinct values the ratio is constant. □

Remark 2 (Small batches). Lemma 12 controls only the first term of Δ_γ, so monotonicity of R_γ does not imply monotonicity of Δ_γ. When the batch size n is small, the selection-noise penalty in (24) cannot be overlooked. Increasing γ spreads the scores b_i further from 1, which typically enlarges this penalty and tightens the uncapped condition $\eta \widehat{a}_i b_i \leq s_G$. A larger γ therefore gains signal at the cost of greater sensitivity to gradient noise.