HESTIA: Hessian-Guided Soft-to-Hard Continuation for Extremely Low-Bit LLM Quantization-Aware Training

Guoan Wang[1,2]*, **Feiyu Wang**[1,3]*, **Zongwei Lv**[1,2], **Yikun Zong**[4], **Zhewen Tan**[1,3], and **Tong Yang**[1,3]†

[1]Beijing Key Laboratory of Software and Hardware Cooperative Artificial Intelligence Systems, Peking University
[2]School of Software and Microelectronics, Peking University
[3]School of Computer Science, Peking University
[4]Department of Engineering, University of Cambridge

gawang25@stu.pku.edu.cn,
{wangfeiyu, yangtong}@pku.edu.cn

Abstract

Quantization-aware training (QAT) is a key approach for adapting large language models (LLMs) to extremely low-bit weights, where post-training quantization often suffers severe accuracy loss. However, most extremely low-bit QAT methods rely on the straight-through estimator (STE), which keeps hard round-and-clip weights in the forward pass while using an identity surrogate in the backward pass, creating a mismatch between discrete forward states and continuous latent-weight updates. As a result, small updates may fail to cross quantization boundaries, causing dead-zone stagnation in sub-optimal discrete states. To address this problem, we propose HESTIA, an operator-faithful differentiable QAT framework for extremely low-bit LLMs based on soft-to-hard continuation. Instead of repairing the gradient of a hard quantizer, HESTIA relaxes the quantizer itself as a temperature-controlled `Softmax` expectation over the same discrete codebook. With detached group scales, the relaxation provides forward–backward consistent gradients during training and uses the target hard quantizer at the endpoint for inference. To stabilize this soft-to-hard transition, HESTIA further uses offline Hessian traces as lightweight tensor-wise sensitivity signals for adaptive temperature scheduling. Experiments on Llama-3.2 show that HESTIA improves average zero-shot accuracy by 5.39% and 4.34% relative to reported Tequila results on 1B and 3B models, with about 2.5% and 2.9% training-time overhead, respectively, including about 0.2% for calibration, and no additional inference-time operations. Example code is available at https://github.com/PKU-LLM-DS-LAB/Hestia.

1 Introduction

Quantization-aware training (QAT) adapts pretrained large language models (LLMs) to low-precision weights by exposing them to quantization during optimization. At extremely low bit-widths, this adaptation is particularly useful because post-training quantization can incur substantial accuracy loss [4, 32, 26]. Most ternary QAT methods use the straight-through estimator (STE) [2]: the forward pass rounds and clips continuous latent weights, while the backward pass bypasses this discrete map.

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

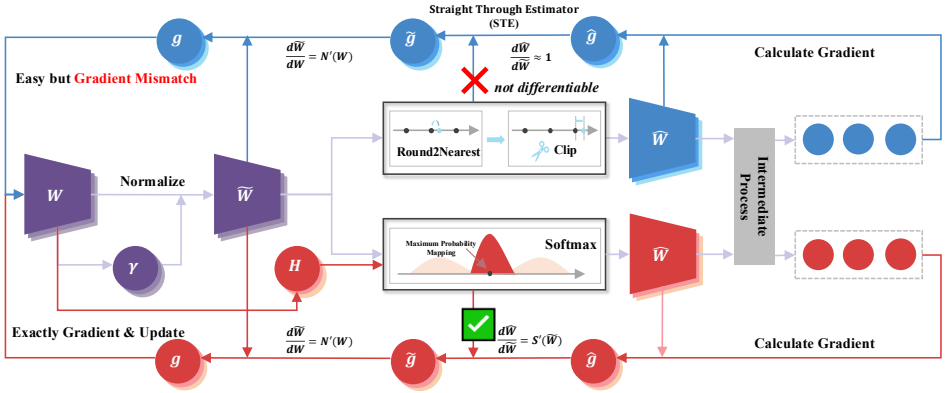


Figure 1: STE-based QAT (top) and HESTIA (bottom). W, $\widetilde{W}$, and $\widehat{W}$ denote latent, normalized, and quantized weights; g, $\widetilde{g}$, and $\widehat{g}$ denote their gradients. During continuation, HESTIA differentiates the same soft weight map used in the forward pass, conditional on a detached group scale.

A nonzero latent update may therefore leave the hard forward weight unchanged until it crosses a quantization boundary. This no-response behavior, discussed as a dead zone in Tequila [15], motivates studying how the training quantizer shapes the optimization path.

We introduce **HESTIA**, a Hessian-guided soft-to-hard continuation framework for extremely low-bit LLM QAT. Its training weights interpolate between latent weights and a temperature-controlled expectation over the target codebook. Pressure controls when quantization enters the forward computation; temperature controls how assignments harden. Offline tensor-wise sensitivity scores modulate the annealing schedule, while all tensors retain their prescribed inference codebook and scale rule. At the endpoint, an explicit hard branch applies the target quantizer, including its tie-breaking convention.

The distance-based Softmax assignment is an established relaxation: in the scalar, fixed-codebook case it is the same primitive used by Agustsson et al. [1]. Pressure interpolation also has precedent in alpha-blending [24]. Our contribution is the specific combination for LLM QAT with a fixed hard endpoint, its sensitivity-conditioned schedule, and an empirical study of the resulting boundary dynamics.

For detached group scales, the relaxed quantizer has Jacobian $(2/\tau)\,\mathrm{Var}_{\pi_\tau}(q)$. It emphasizes uncertain code assignments and concentrates near decision boundaries as temperature falls. The loss gradient still determines the update direction. Empirically, across 973,078,528 examined weights, the fraction within normalized distance 0.1 of a ternary boundary falls from 11.84% in the base checkpoint to 3.66% under HESTIA, whereas hard-STE training raises it to 23.66%. This separation occurs without the auxiliary penalty used by Bin Regularization [14]. Short and terminal diagnostics show that finite-temperature responsiveness eventually gives way to near-freezing as intended by hardening.

Our main contributions are: (i) a coupled pressure-and-temperature continuation recipe with a fixed inference endpoint; (ii) a variance-Jacobian analysis with explicit scale and boundary assumptions, linked to direct response and boundary measurements; and (iii) controlled Llama-3.2-1B comparisons and 1B/3B scheduling ablations. The controlled comparison yields average accuracy 0.547 versus 0.530 for Tequila and 0.526 for DSQ-style QAT. Hessian scheduling improves perplexity and several panel averages over a global schedule and reaches average accuracy 0.547 compared with 0.543 for GradNorm. Nonuniform-codebook and MoE studies further examine the transfer of the continuation framework.

2 Related Work

Extremely low-bit LLM QAT. BitNet, Spectra, ParetoQ, and related methods study low-bit architectures and training recipes [35, 26, 18, 23, 9]. TWN and AbsMean use hard ternary weights [21]; Tequila reactivates dead weights [15]. PV-Tuning and CAGE improve discrete optimization through reparameterization and curvature-aware gradient estimation [27, 31]. These approaches motivate examining the forward operator and its training path, alongside data and model scaling.

Soft assignments and continuation. Agustsson et al. [1] use distance-based Softmax assignments and anneal them for compressible representations, including network weights. Our scalar fixed-codebook assignment is mathematically shared with that construction. Louizos et al. [25] study probabilistic relaxed quantization, and Liu and Mattina [24] interpolate full-precision and quantized weights through alpha-blending. DSQ [13] uses interval-wise smooth approximations with a hard forward mapping. DAQ and more recent LLM methods explore other differentiable approximations [19, 36, 20]. HESTIA combines a soft expectation in the forward pass with pressure interpolation and tensor-wise annealing; these components and their antecedents are compared in Appendix F. Boundary separation is also an explicit goal of Bin Regularization [14], which adds a penalty to hard-STE training. Our experiments measure boundary separation induced by the continuation trajectory without adding a bin penalty.

Sensitivity-guided quantization. HAWQ uses curvature for mixed-precision allocation [8]; QBitOpt reallocates bit-widths during training using sensitivity information [29]; QuIP# uses incoherence processing and lattice codebooks for accurate low-bit quantization [32]. HESTIA retains each tensor's target codebook and bit-width and instead controls the temperature of its continuation path. This is a specific use of established sensitivity-guided design. Our GradNorm ablation tests a cheaper first-order proxy and prevents attributing the full continuation gain uniquely to second-order information.

3 Method

3.1 Problem Definition

We first present the main ternary setting, and later extend the same expectation-based relaxation to other finite target codebooks by replacing the support set $\mathcal{Q}$. Consider a weight matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$ constrained to ternary codes $\mathcal{Q} = \{-1, 0, +1\}$. For clarity, we write the quantizer for one tensor or quantization group; in grouped quantization, the same equations apply independently within each group. We use a scaled element-wise quantizer with

$$\gamma = \frac{1}{N}\sum_{j=1}^{N} |w_j| + \epsilon_\gamma, \tag{1}$$

where N is the group size, $w_j \in \mathbf{W}$, $\epsilon_\gamma = 10^{-8}$, and γ is recomputed in the forward pass but detached during backpropagation. All derivative statements below condition on this detached scale. The dequantized weights are

$$\hat{\mathbf{W}} = Q(\mathbf{W}) = \gamma \cdot \texttt{Clip}\left(\texttt{Round}\left(\frac{\mathbf{W}}{\gamma}\right), -1, +1\right), \tag{2}$$

The empirical objective over $\mathcal{D}$ is

$$\min_{\mathbf{W}} \mathcal{J}(\mathbf{W}), \text{where} \mathcal{J}(\mathbf{W}) = \mathbb{E}_{x\sim\mathcal{D}}[\mathcal{L}(Q(\mathbf{W}); x)]. \tag{3}$$

With the scale held fixed, $Q(\cdot)$ is piecewise constant, so standard QAT uses STE and treats quantization as an identity map in the backward pass:

$$\nabla_{\mathbf{W}}\mathcal{J} = \nabla_{\hat{\mathbf{W}}}\mathcal{L} \cdot \frac{\partial \hat{\mathbf{W}}}{\partial \mathbf{W}} \approx \nabla_{\hat{\mathbf{W}}}\mathcal{L}. \tag{4}$$

STE enables training, but mismatches continuous latent updates and discrete state transitions. For the clipped ternary operator, the transition boundaries are

$$\mathcal{B} = \{-0.5\gamma,\ +0.5\gamma\}, \tag{5}$$

because `Clip` removes output changes beyond $|w/\gamma| \geq 1.5$. For a latent update Δw, with the same frozen scale in both evaluations, define the no-response set as

$$\Omega_{\text{dead}} = \{w \in \mathbf{W} \mid Q(w + \Delta w) = Q(w)\}. \tag{6}$$

For $w \in \Omega_{\text{dead}}$, this update leaves the hard forward value unchanged. Successive latent updates can accumulate until a later boundary crossing.

3.2 Differentiable Quantization

We replace $Q(\cdot)$ with a differentiable surrogate $\mathcal{H}(\cdot;\tau)$ that gradually transitions from soft to hard assignments. For each latent weight w, the assignment to $q \in \mathcal{Q}$ is given by

$$\pi_\tau(q \mid w) = \texttt{Softmax}(w; q; \tau) = \frac{e^{-(w/\gamma - q)^2/\tau}}{\sum_{k\in\mathcal{Q}} e^{-(w/\gamma - k)^2/\tau}}, \tag{7}$$

The relaxed weight is the code expectation

$$\tilde{w} = \mathcal{H}(w;\tau) = \gamma \sum_{q\in\mathcal{Q}} q \cdot \pi_\tau(q \mid w). \tag{8}$$

For a unique nearest code, $\mathcal{H}(w;\tau) \to Q(w)$ as $\tau \to 0^+$. At an exact tie, the limit averages the tied codes and need not equal deterministic rounding. We therefore define the endpoint separately: at $t = T$, set $p_T = 1$ and apply the hard quantizer with the same tie rule as inference, without evaluating a division by zero. For real nonuniform codebooks, replace $\mathcal{Q}$ by the actual target levels and use nearest-code hard quantization; Appendix H discusses NF4 and complex codes. The Softmax primitive itself follows prior soft-to-hard quantization [1].

To avoid projecting weights too abruptly early in training, we introduce a compress-stage pressure $p_t \in [0,1]$ and use

$$\mathbf{W}^{\text{eff}}(t) = (1 - p_t) \cdot \mathbf{W} + p_t \cdot \mathcal{H}(\mathbf{W};\tau), \tag{9}$$

where

$$p_t = \begin{cases} 1, & \rho = 0, \\ \min\left(1, \frac{t}{\rho T}\right), & \rho \in (0,1), \end{cases} \tag{10}$$

t is the current step, T is the total number of steps, and $\rho \in [0,1)$ is the compress-stage fraction. This continuation path separates when quantization is introduced from how assignments harden. Training optimizes

$$\tilde{\mathcal{J}}_t(\mathbf{W}) = \mathbb{E}_{x\sim\mathcal{D}}[\mathcal{L}(\mathbf{W}_{\text{eff}}(t); x)], \tag{11}$$

which approaches the discrete objective away from ties and uses it explicitly at the hard endpoint. Both p_t and $\tau_i(t)$ are prescribed schedule values during differentiation.

3.3 Hessian-Guided Tensor-Wise Annealing

While $\mathcal{H}(\cdot;\tau)$ provides gradient flow, a uniform annealing schedule ignores the structural heterogeneity of LLM tensors. HESTIA therefore assigns each weight tensor its own annealing rate based on quantization sensitivity.

Sensitivity Estimation. We estimate each tensor's sensitivity with Hessian trace, approximated efficiently by Hutch++ [28]. The trace is computed once from the full-precision calibration loss, before applying the hard quantizer, so it serves as a curvature prior rather than a Hessian obtained by differentiating through the hard assignments in Eq. (3). For weight tensor i, let $\mathbf{w}_i = \text{vec}(\mathbf{W}_i)$ and $\mathbf{H}_i$ denote the Hessian of this calibration loss with respect to $\mathbf{w}_i$. Hutch++ estimates

$$h_i = \text{Tr}(\mathbf{Q}_i^\top \mathbf{H}_i \mathbf{Q}_i) + \frac{1}{m}\text{Tr}\big(\mathbf{G}_i^\top (\mathbf{I} - \mathbf{Q}_i\mathbf{Q}_i^\top)\mathbf{H}_i(\mathbf{I} - \mathbf{Q}_i\mathbf{Q}_i^\top)\mathbf{G}_i\big), \tag{12}$$

where $\mathbf{Q}_i$ is an orthonormal sketched basis and $\mathbf{G}_i$ has m independent, unscaled Rademacher columns. The residual average requires the factor $1/m$. Define $\ell_i = \log\max(h_i, \epsilon_h)$, with $\epsilon_h = 10^{-8}$, and normalize

$$s_i = \texttt{Sigmoid}\left(\kappa \cdot \frac{\ell_i - \mu_\ell}{\sigma_\ell + \epsilon_h}\right), \tag{13}$$

where μ_ℓ and σ_ℓ are the mean and standard deviation across calibrated linear tensors, and $\kappa = 1$. These are relative scheduling scores, not certified error bounds. A neural-network Hessian can be indefinite and its trace estimate noisy; flooring nonpositive estimates makes the transform defined but discards their signed information. Appendix C compares this prior with GradNorm.

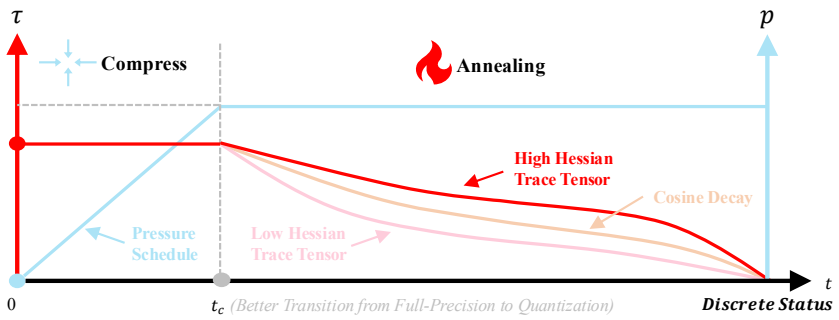


Figure 2: The HESTIA adaptive annealing schedule. During the compress stage, pressure p increases linearly while temperature remains fixed; during annealing, tensor-wise temperatures decay according to Hessian-derived sensitivity.

Temperature Annealing Schedule. Given s_i, we define a base schedule that stays fixed during compression and then follows a cosine decay:

$$\bar{\tau}(t) = \begin{cases} \tau_{\text{init}}, & t \le T_{\text{comp}}, \\ \frac{\tau_{\text{init}}}{2}\left[1 + \cos\left(\frac{\pi(t - T_{\text{comp}})}{T - T_{\text{comp}}}\right)\right], & t > T_{\text{comp}}, \end{cases} \tag{14}$$

where $T_{\text{comp}} = \rho T$ is the end of the compress stage, and assign each tensor

$$\tau_i(t) = \bar{\tau}(t) \cdot e^{\alpha s_i}. \tag{15}$$

For $t < T$, larger s_i retains a higher temperature, while all tensors use the same hard endpoint. Scores are computed once and held fixed. For $\alpha \ge 0$, $\max_i \tau_i / \min_i \tau_i \le e^{\alpha}$ (about 1.49 at $\alpha = 0.4$). This bounds the temperature spread induced by the fixed sensitivity prior. The global variant sets $\alpha = 0$; GradNorm substitutes a gradient-norm signal in the same log-standardization and temperature map.

4 Theoretical Analysis

Building on the relaxed quantizer $\tilde{w} = \mathcal{H}(w; \tau)$ in Eq. (8), we study how the annealed assignment kernel $\pi_\tau(q \mid w)$ in Eq. (7) shapes backpropagation. Specifically, we prove (i) a closed-form Jacobian showing that gradients are modulated by assignment uncertainty, (ii) boundary localization as $\tau \to 0^+$ at the hard quantization boundaries $\mathcal{B}$ in Eq. (5), and (iii) an exponentially decreasing soft-to-hard error away from ties.

Throughout, $\gamma > 0$ is fixed during differentiation and $\tau > 0$. Lemma 4.1 holds for any finite real codebook containing at least two distinct values; Lemma 4.2 specializes to ternary codes. We denote the code mean under $\pi_\tau(\cdot \mid w)$ by

$$\mu_\tau(w) \triangleq \sum_{q \in \mathcal{Q}} q\, \pi_\tau(q \mid w), \tag{16}$$

and the corresponding variance by

$$\mathbb{V}_\tau(w) \triangleq \sum_{q \in \mathcal{Q}} q^2\, \pi_\tau(q \mid w) - \mu_\tau(w)^2. \tag{17}$$

We give the key proof steps below; complete derivations appear in Appendix B.

Lemma 4.1 (Variance-Modulated Jacobian)**.** *For $\tilde{w} = \mathcal{H}(w; \tau)$, the Jacobian w.r.t. the latent weight satisfies*

$$\frac{\partial \mathcal{H}(w; \tau)}{\partial w} = \frac{2}{\tau}\, \mathbb{V}_\tau(w). \tag{18}$$

Proof sketch. Set $z = w/\gamma$. Differentiating the normalized assignment gives $\partial_z \pi_\tau(q \mid z) = (2/\tau)\pi_\tau(q \mid z)(q - \mu_\tau)$, as derived in Eq. (27). Multiplying by q and summing over the codebook yields $\partial_z \mu_\tau = (2/\tau)(\mathbb{E}_\tau[q^2] - \mu_\tau^2)$. The scale factors in $\mathcal{H} = \gamma\mu_\tau$ and $\partial_w z = 1/\gamma$ cancel, giving Eq. (18). This argument uses only a finite codebook and a fixed positive scale, so the same identity applies to nonuniform real levels.

Lemma 4.1 implies an uncertainty-gated backward signal: for any scalar loss $\ell(\tilde{w})$,

$$\frac{\partial \ell(\mathcal{H}(w;\tau))}{\partial w} = \frac{\partial \ell}{\partial \tilde{w}} \cdot \frac{2}{\tau}\, \mathbb{V}_\tau(w), \tag{19}$$

so gradients are attenuated when $\pi_\tau(\cdot \mid w)$ is confident and amplified when assignments are ambiguous.

Lemma 4.2 (Boundary Localization as $\tau \to 0^+$)**.** *As $\tau \to 0^+$, the Jacobian $\partial\mathcal{H}(w;\tau)/\partial w$ converges in the sense of distributions to Dirac measures supported on $\mathcal{B}$:*

$$\frac{\partial \mathcal{H}(w;\tau)}{\partial w} \xrightarrow[\tau \to 0^+]{} \gamma \sum_{b \in \mathcal{B}} \delta(w - b). \tag{20}$$

Proof sketch. Away from a tie, the ratio of a non-nearest assignment to the nearest assignment is $e^{-\Delta_q/\tau}$ for a positive squared-distance gap Δ_q (Eq. (31)). Thus $\mathcal{H}(w;\tau) \to Q(w)$ almost everywhere. Since $|\mathcal{H}| \le \gamma$, integration by parts and dominated convergence give, for every smooth compactly supported test function φ,

$$\lim_{\tau \to 0^+} \int \partial_w \mathcal{H}(w;\tau)\varphi(w)\, dw = -\int Q(w)\varphi'(w)\, dw = \gamma \sum_{b \in \mathcal{B}} \varphi(b). \tag{21}$$

The final equality follows from the two jumps of height γ in the clipped ternary map. Each boundary therefore retains its jump mass as the smooth transition narrows; rounding transitions removed by clipping contribute no mass.

Lemma 4.2 formalizes a localization effect: annealing progressively concentrates the surrogate Jacobian near decision boundaries, turning smooth continuation into near-discrete refinement. For the pressure-interpolated operator, the conditional Jacobian is $(1 - p_t) + p_t(2/\tau)\mathbb{V}_\tau(w)$. At finite temperature it is positive in exact arithmetic, but can be numerically negligible or vanish through finite-precision saturation. The loss gradient determines the update direction, while the variance factor modulates its magnitude. Section 5.5 measures the resulting boundary distributions along the training trajectories.

Proposition 4.3 (Soft-to-hard error away from ties)**.** *Let $q^\star$ be the unique nearest code to $z = w/\gamma$ in a finite real codebook of size K, and let $\Delta = \min_{q \neq q^\star}[(z-q)^2 - (z-q^\star)^2] > 0$. Then*

$$|\mathcal{H}(w;\tau) - \gamma q^\star| \le \gamma \,\mathrm{diam}(\mathcal{Q})(K-1)e^{-\Delta/\tau}. \tag{22}$$

Proof sketch. The probability ratio above implies $\pi_\tau(q \mid w) \le e^{-\Delta/\tau}$ for every $q \neq q^\star$. Their combined mass is at most $(K-1)e^{-\Delta/\tau}$. Multiplying by the largest possible displacement, $\gamma\,\mathrm{diam}(\mathcal{Q})$, bounds the difference between the code expectation and $\gamma q^\star$.

The approximation tightens exponentially with Δ/τ away from ties; at decision boundaries, Δ vanishes. Our experiments measure the corresponding finite-temperature response, boundary mass, and loss at the hard endpoint.

5 Experiments

5.1 Experimental Setup

We evaluate HESTIA on official `Llama-3.2-1B/3B` checkpoints [10], with a nominal QAT budget of 10B Ultra-FineWeb tokens [37]. Transformer linear weights use ternary group-size-128 quantization, while activations remain in full precision. We use AdamW with learning rate 5×10^{-5}, weight decay

0.01, and global batch size 256. Offline calibration uses batch size 50, sketch rank 10, and 20 Hutchinson samples. Appendix A lists the full configuration. Accuracy uses `lm-evaluation-harness` [12] on ARC-Easy/Challenge [6], HellaSwag [38], PIQA [3], and WinoGrande [30].

The controlled 1B comparison matches the checkpoint, data, nominal budget, optimizer, group size, and evaluation protocol. DSQ-style QAT adapts the published interval-wise tanh relaxation and hard-forward mapping to the same ternary scaling and clipping rule. Global, GradNorm, and Hessian variants share the continuation recipe but use different temperature signals. The main HESTIA results are means over five runs; the additional controlled baselines and ablations are point estimates. Appendix I summarizes reporting conventions and code availability; the shorter diagnostic runs are described separately from the main budget.

5.2 Controlled Comparison and Generalization

Table 1 gives the main controlled comparison. The global Softmax continuation gains 5.1 percentage points over STE/AbsMean, reflecting the combined operator and pressure path. Full HESTIA exceeds controlled Tequila by 1.7 points on average, improving all five tasks; it exceeds DSQ-style QAT by 2.1 points, improving four. Hessian and GradNorm scores are strongly rank-correlated ($r_s = 0.973$). Their downstream averages are 0.547 and 0.543, respectively, placing GradNorm 0.4 percentage points below Hessian scheduling while offering a lower-cost sensitivity signal. Heterogeneous published baselines are presented separately in Appendix E. Where highlighted, gray denotes BF16 references and red/blue mark the largest/second-largest accuracy (or lowest/second-lowest perplexity) within the indicated comparison.

Table 1: Controlled Llama-3.2-1B ternary comparison. †: five-run mean; other rows: point estimates. Avg. is the five-task accuracy average.

Method	ARC-E	ARC-C	HellaSwag	PIQA	WinoGrande	Avg.
STE/AbsMean	.603	.259	.360	.683	.541	.489
DSQ-style	.604	.330	.391	.718	.588	.526
Tequila (controlled)	.628	.328	.396	.724	.573	.530
Softmax + global	.654	.323	.422	.720	.580	.540
Softmax + GradNorm	.651	.349	.416	.727	.571	.543
HESTIA†	.665	.350	.418	.726	.577	.547

Other codebooks and domains. Table 2 reports a six-task average including GPQA. HESTIA obtains 0.461 at 1-bit, 0.494 at ternary precision, and 0.490 at 2-bit on Llama-3.2-1B; quality is not monotone in bit-width for these runs. Table 3 applies the continuation to real ternary and complex Fairy2i pipelines [33, 34] on Qwen2.5-0.5B. Average accuracy increases from 0.437 to 0.462 in the real ternary setting and from 0.471 to 0.474 in the complex setting. Section 5.6 examines NF4 endpoint compatibility; Appendix H reports untuned MoE transfer.

Table 2: Generalization across different target bit-widths on `Llama-3.2-1B`.

Method	Bit	ARC-E	ARC-C	HellaSwag	PIQA	WinoGrande	GPQA	Avg. (↑)
BF16	BF16	**0.654**	**0.313**	**0.477**	**0.743**	**0.603**	**0.222**	**0.502**
HESTIA	1-bit	0.638	0.279	0.380	0.703	0.543	0.222	0.461
AbsMean	1.58-bit	0.603	0.259	0.360	0.683	0.541	0.227	0.446
Tequila	1.58-bit	0.645	0.305	0.391	0.710	0.542	**0.232**	0.471
HESTIA	1.58-bit	**0.665**	**0.350**	**0.418**	**0.726**	**0.577**	**0.227**	**0.494**
AbsMean	2-bit	0.619	0.262	0.354	0.692	0.527	**0.227**	0.447
HESTIA	2-bit	**0.669**	**0.331**	**0.429**	**0.723**	**0.578**	0.212	**0.490**

5.3 Ablation on Hessian-Guided Scheduling

Comparing scheduling signals. Figure 3 compares the offline Hessian trace, gradient norm, and activation variance. Hessian trace and gradient norm give closely related tensor rankings ($r_s = 0.973$), while Hessian trace spans a wider dynamic range in the calibration distribution. The controlled results in Table 1 compare these signals under the same continuation recipe. Both use scores computed once before QAT. Appendix C further shows how traces vary across projection types and layers.

Table 3: Cross-domain results on `Qwen2.5-0.5B`. Baseline is the matched hard-STE pipeline for real ternary weights, and the original Fairy2i optimization pipeline for complex weights, using the same checkpoint and evaluation protocol.

Method	Domain	Training	ARC-E	ARC-C	HellaSwag	PIQA	WinoGrande	Avg. (↑)
BF16	**Real**	–	**0.645**	**0.290**	**0.405**	**0.705**	**0.567**	**0.523**
Ternary	Real	Baseline	0.498	0.207	0.321	0.649	0.509	0.437
		+HESTIA	0.539	0.221	0.349	0.667	0.532	0.462
Fairy2i	Complex	Baseline	0.540	0.238	0.357	0.678	0.540	0.471
		+HESTIA	0.556	0.236	0.357	0.670	0.549	0.474

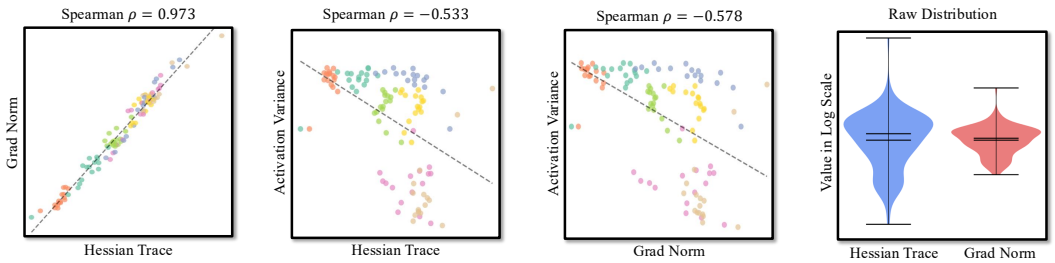


Figure 3: Comparison of candidate scheduler signals. Hessian trace and gradient norm give closely related tensor rankings, but Hessian trace provides a wider dynamic range, while activation variance is less discriminative across several projection types. Downstream scheduling results are compared in Table 1.

Table 4: Global versus Hessian scheduling. W/o means global temperature ($\alpha = 0$), with the same soft continuation; w/ means Hessian guidance.

(a) Common-sense reasoning accuracy and validation perplexity.

		Common-sense reasoning						Validation perplexity		
Hessian	Size	ARC-E	ARC-C	HellaSwag	PIQA	WinoGrande	Avg. (↑)	WikiText	C4	Avg. (↓)
BF16	**1B**	**0.654**	**0.313**	**0.477**	**0.743**	**0.603**	**0.558**	**11.413**	**14.219**	**12.816**
w/o	1B	0.654	0.323	**0.422**	0.720	**0.580**	0.540	27.034	21.681	24.358
w/		**0.665**	**0.350**	0.418	**0.726**	0.577	**0.547**	24.452	**21.538**	**22.995**
BF16	**3B**	**0.745**	**0.422**	**0.552**	**0.768**	**0.691**	**0.636**	**9.227**	**10.528**	**9.877**
w/o	3B	0.738	**0.402**	0.487	0.753	0.612	0.598	20.627	18.377	19.502
w/		**0.741**	0.393	**0.491**	**0.757**	**0.622**	**0.601**	**18.547**	**17.271**	**17.909**

(b) Extended knowledge, math, and reading comprehension benchmarks.

Hessian	Size	GPQA	MMLU	GSM8K	DROP	AGIEval	BoolQ	OpenBookQA	Avg. (↑)
BF16	**1B**	**0.222**	**0.322**	**0.075**	**0.027**	**0.176**	**0.637**	**0.278**	**0.248**
w/o	1B	**0.247**	**0.258**	0.015	0.047	0.164	0.429	0.130	0.184
w/		0.227	0.255	**0.026**	0.047	**0.179**	**0.563**	**0.248**	**0.221**
BF16	**3B**	**0.303**	**0.549**	**0.296**	**0.016**	**0.239**	**0.736**	**0.314**	**0.350**
w/o	3B	**0.293**	0.295	0.069	0.049	0.184	0.619	0.265	0.253
w/		0.263	**0.304**	**0.073**	**0.060**	**0.187**	**0.678**	**0.278**	**0.263**

We compare HESTIA with the same continuation using a global schedule ($\alpha = 0$), keeping $(\rho, \text{group size}) = (0.2, 128)$. Table 4 shows improvements in each panel average at both scales. The 1B global WikiText perplexity is 27.034, giving a two-corpus mean of 24.358 versus 22.995 for Hessian scheduling; the corresponding 3B mean decreases from 19.502 to 17.909. On 1B, the extended average rises from 0.184 to 0.221, mainly through BoolQ and OpenBookQA. GPQA decreases from 0.247 to 0.227, with both values near its four-choice chance level of 0.25. The clearest scheduling gains occur in perplexity, BoolQ, and OpenBookQA.

5.4 Sensitivity to Schedule Hyperparameters

We sweep the temperature-scaling strength α and compress-stage ratio ρ around the default $(0.4, 0.2)$, changing one variable at a time in Table 5. We additionally isolate quantization group size under the

Table 5: Sensitivity of HESTIA to schedule hyperparameters on `Llama-3.2-1B/3B`. Bold entries mark the default value for each swept hyperparameter.

Hyperparameter	Size	Value	ARC-E	ARC-C	HellaSwag	PIQA	WinoGrande	Avg. (↑)
temperature-scaling strength α	1B	-0.2	0.647	0.324	0.419	**0.726**	**0.580**	0.539
	1B	0.2	**0.665**	0.328	**0.421**	**0.729**	**0.582**	**0.545**
	1B	**0.4**	**0.665**	**0.350**	0.418	**0.726**	0.577	**0.547**
	1B	0.6	**0.659**	**0.332**	**0.423**	**0.726**	0.567	0.542
compress-stage ratio ρ	1B	0.0	0.650	0.311	**0.420**	0.721	0.574	0.535
	1B	0.1	0.657	0.319	**0.420**	0.721	0.571	0.538
	1B	**0.2**	**0.665**	**0.350**	0.418	**0.726**	**0.577**	**0.547**
	1B	0.3	**0.667**	**0.326**	0.417	**0.724**	0.569	**0.541**
group size	1B	—	**0.659**	0.322	0.410	**0.724**	0.571	0.537
	1B	**128**	0.654	**0.323**	**0.422**	0.720	**0.580**	**0.540**
	3B	—	0.731	0.397	0.485	0.746	**0.627**	0.597
	3B	**128**	**0.738**	**0.402**	**0.487**	**0.753**	0.612	**0.598**

global-schedule setting ($\alpha = 0$), so the group-size rows should be compared within that block rather than against the full Hessian-guided default.

In the α sweep, positive values in $[0.2, 0.6]$ outperform $\alpha = -0.2$, consistent with delaying hardening of higher-score tensors in this setting. Accuracy varies by only 0.005 across this positive range, indicating low sensitivity within this tested range.

The ρ and group-size sweeps show similarly mild sensitivity. Among the tested pressure ratios, removing the compress stage gives the lowest average, while $\rho = 0.2$ gives the best balance; $\rho = 0.1$ and 0.3 remain within 0.012 of the optimum. Within the group-size block, group size 128 slightly improves over the ungrouped variant and matches the competing ternary baselines.

Interpreting the defaults. The schedule parameters control distinct aspects of the transition: ρ sets the duration of pressure interpolation, τ_{init} sets the initial assignment softness, and α bounds the temperature spread across tensors. Since $s_i \in (0, 1)$, choosing $\alpha = \log r_{\max}$ bounds this spread by $r_{\max}$; the default $\alpha = 0.4$ allows a ratio of at most 1.49. Initial softness can be read from code spacing: at a code center, a neighbor at distance δ has relative assignment mass $e^{-\delta^2/\tau}$. For unit-spaced ternary codes, the base temperature 0.3 gives approximately 0.036. These quantities provide interpretable starting values when adapting the continuation to another codebook; Appendix H discusses nonuniform spacings.

5.5 Boundary Dynamics

We examine all 973,078,528 weights in 112 projection tensors using normalized $z = w/\gamma$ and margin $m(z) = \min(|z - 0.5|, |z + 0.5|)$. Table 6 shows that HESTIA reduces boundary-adjacent mass without an auxiliary bin regularizer, whereas hard STE increases it. At step 31,000, mass within 0.1 is already 3.67% for HESTIA and 23.56% for STE, so the final separation is not solely a last-step effect. These statistics summarize the full examined weight population at each checkpoint.

Table 6: Distribution of normalized boundary margins over the examined weight population. Fractions are percentages; mean margin is dimensionless.

Checkpoint	$m \leq .01$	$m \leq .05$	$m \leq .10$	Mean m
Base FP	1.185	5.93	11.84	.659
Hard STE	4.925	15.18	23.66	.643
HESTIA	.009	.59	3.66	.690

A separate 100-step diagnostic rescales the schedule to that horizon and tracks 512 fixed weights in each of 21 tensors. Each retained STE-trajectory update is applied to both maps, holding the group scale, pressure, and temperature fixed between evaluations. This isolates the forward response to the latent update. We retain $|\Delta w| > 10^{-7}$ and count a response when the effective weight changes by more than 10^{-12}. Table 7 reports the event counts and intervals. The no-response fraction for the HESTIA map is 0%, 18.30%, and 86.45% in early, middle, and late windows, versus over 99.95% for the hard map. In the final 690 updates of a separate 8.307B-token trajectory, response is only 0.00044% for HESTIA and 0% for the hard map. Together, these diagnostics characterize

early/mid-stage responsiveness and terminal near-freezing. Appendix G supplies window definitions and additional trajectory and threshold checks.

Table 7: No-response fractions on STE-trajectory updates in the 100-step diagnostic. Brackets are 95% intervals obtained by bootstrapping layer-step clusters. Numerators count no-response events.

Window	Retained updates	Hard no response	HESTIA no response	95% intervals (hard; HESTIA)
Early	106,954	106,906 (99.955%)	0 (0%)	[99.943,99.966]; [0,0]
Middle	106,652	106,602 (99.953%)	19,514 (18.30%)	[99.940,99.965]; [17.93,18.66]
Late	106,517	106,486 (99.971%)	92,087 (86.45%)	[99.959,99.982]; [85.50,87.41]

The measured Jacobian also follows the localization in Lemma 4.2: at step 31,000 and $\tau = 0.01$, every measured distance bin beyond normalized margin 0.1 has mean absolute Jacobian below 6×10^{-9}. Thus, late-stage responsiveness is concentrated near competing assignments, while weights farther from a boundary have nearly fixed effective values.

5.6 Nonuniform Codebook and Endpoint Validation

For ordered real levels $\mathcal{Q} = \{q_1 < \cdots < q_K\}$, the expectation and variance Jacobian retain the same form. Nearest-code boundaries lie at $(q_j + q_{j+1})/2$ in normalized coordinates, with hard output jumps of $\gamma(q_{j+1} - q_j)$. Uniform spacing is therefore unnecessary. At an exact tie, the soft limit averages the tied levels; the explicit hard branch reproduces the target quantizer's deterministic tie convention. These properties allow the training relaxation to use the target codebook directly.

We further test the fixed-codebook construction with NF4, whose 16 nonuniform levels were introduced with QLoRA [7]. Using group size 64, absmax scaling, and nearest-code lookup, the hard endpoint matches bitsandbytes bit-for-bit over 1,048,576 tested values. A 2,000-step Qwen2.5-0.5B continuation processes 512,000 packed sequences, approximately 524M token slots, with a disjoint 256-sequence validation set. Final scheduled and forced-hard validation NLL are both 3.26766 (perplexity 26.25), and the weight mean-squared difference is zero. This directly checks that the learned endpoint uses the target discrete values. The experiment extends the continuation beyond uniform ternary spacing while retaining the prescribed quantization format; Appendix H gives further details.

5.7 Computational Efficiency

For group size 128, measured training cost is 419.9 versus 409.6 GPU-hours at 1B and 1477.9 versus 1435.8 GPU-hours at 3B on RTX 4090 GPUs (24GB): approximately 2.5% and 2.9% overhead relative to the matched ternary baseline. One-time Hutch++ calibration takes 0.9 and 2.7 GPU-hours, respectively, about 0.2% of total HESTIA time. The remaining overhead arises from relaxed-quantizer operations during training. Calibration stores one scalar sensitivity score per tensor, which is reused throughout QAT (Appendix D). At inference, the continuation adds no operation to the chosen hard grouped-ternary format. Deployment latency and memory savings depend on weight packing and engine/kernel support.

6 Conclusion and Limitations

HESTIA combines soft assignments with pressure interpolation and Hessian-guided annealing into a fixed-endpoint continuation for extremely low-bit LLM QAT. The variance Jacobian characterizes boundary-sensitive gradients, and direct measurements show boundary separation without an auxiliary bin penalty. Controlled comparisons yield average accuracy 0.547 on Llama-3.2-1B, exceeding Tequila by 1.7 percentage points and DSQ-style QAT by 2.1 points. Hessian scheduling further improves perplexity and selected task averages over a global schedule. Total training overhead is approximately 2.5%/2.9% at 1B/3B, with the same grouped-ternary inference format.

The main evaluation covers weight-only dense models up to 3B and uses a fixed calibration prior. Larger models, dynamic sensitivity updates, and joint weight–activation quantization are directions for further study. NF4 and MoE experiments demonstrate initial transfer: the untuned MoE configuration improves over hard STE while retaining a substantial gap to BF16. Load-aware calibration, routing-aware schedules, and deployment benchmarks can further develop these extensions.

Acknowledgments and Disclosure of Funding

This work was supported by the National Key Research and Development Program of China under Grant No. 2024YFB2906602, and in part by the National Natural Science Foundation of China (NSFC) (No. 62372009). The authors declare no other competing interests relevant to this work during the 36 months preceding submission.

References

[1] Eirikur Agustsson, Fabian Mentzer, Michael Tschannen, Lukas Cavigelli, Radu Timofte, Luca Benini, and Luc Van Gool. Soft-to-hard vector quantization for end-to-end learning compressible representations. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

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

[3] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020.

[4] Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M De Sa. QuIP: 2-bit quantization of large language models with guarantees. *Advances in Neural Information Processing Systems*, 36:4396–4429, 2023.

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

[6] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge, 2018. URL `https://arxiv.org/abs/1803.05457`.

[7] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. *arXiv preprint arXiv:2305.14314*, 2023.

[8] Zhen Dong et al. HAWQ: Hessian-aware quantization of neural networks with mixed-precision. *ICCV*, 2019.

[9] Aleksandr Dremov, David Grangier, Angelos Katharopoulos, and Awni Hannun. Compute-optimal quantization-aware training. *arXiv preprint arXiv:2509.22935*, 2025. Apple Machine Learning Research.

[10] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

[11] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization, 2020. URL `https://arxiv.org/abs/1902.08153`.

[12] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL `https://zenodo.org/records/12608602`.

[13] Ruihao Gong, Xianglong Liu, Shenghu Jiang, Tianxiang Li, Peng Hu, Jiazhen Lin, Fengwei Yu, and Junjie Yan. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *ICCV*, 2019.

[14] Tiantian Han, Dong Li, Ji Liu, Lu Tian, and Yi Shan. Improving low-precision network quantization via bin regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5261–5270, 2021.

[15] Hong Huang, Decheng Wu, Rui Cen, Guanghua Yu, Zonghang Li, Kai Liu, Jianchen Zhu, Peng Chen, Xue Liu, and Dapeng Wu. Tequila: Trapping-free ternary quantization for large language models. In *International Conference on Learning Representations*, 2026.

[16] Hong Huang, Decheng Wu, Qiangqiang Hu, Guanghua Yu, Jinhai Yang, Jianchen Zhu, Xue Liu, and Dapeng Wu. Sherry: Hardware-efficient 1.25-bit ternary quantization via fine-grained sparsification. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, 2026.

[17] IBM Granite Team. Granite-3.1-1B-A400M-Base model card. `https://huggingface.co/ibm-granite/granite-3.1-1b-a400m-base`, 2024. Released December 18, 2024.

[18] Ayush Kaushal, Tejas Vaidhya, Arnab Kumar Mondal, Tejas Pandey, Aaryan Bhagat, and Irina Rish. Spectra: Surprising effectiveness of pretraining ternary language models at scale. *arXiv preprint arXiv:2407.12327*, 2024.

[19] Dohyung Kim et al. Distance-aware quantization. In *ICCV*, 2021.

[20] Mujin Kwun, Depen Morwani, Chloe Huangyuan Su, Stephanie Gil, Nikhil Anand, and Sham Kakade. LOTION: Smoothing the optimization landscape for quantized training, 2025. URL `https://arxiv.org/abs/2510.08757`.

[21] Fengfu Li, Bin Liu, Xiaoxing Wang, Bo Zhang, and Junchi Yan. Ternary Weight Networks. *arXiv preprint arXiv:1605.04711*, 2016.

[22] 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. *arXiv preprint arXiv:2305.17888*, 2023.

[23] Zechun Liu, Changsheng Zhao, Hanxian Huang, Sijia Chen, Jing Zhang, Jiawei Zhao, Scott Roy, Lisa Jin, Yunyang Xiong, Yangyang Shi, et al. ParetoQ: Scaling laws in extremely low-bit LLM quantization. *arXiv preprint arXiv:2502.02631*, 2025.

[24] Zhi-Gang Liu and Matthew Mattina. Learning low-precision neural networks without straight-through estimator (STE). In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2019.

[25] Christos Louizos, Matthias Reisser, Tijmen Blankevoort, Efstratios Gavves, and Max Welling. Relaxed quantization for discretized neural networks. In *International Conference on Learning Representations*, 2019.

[26] Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Lifeng Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit LLMs: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*, 1(4), 2024.

[27] Vladimir Malinovskii, Denis Mazur, Ivan Ilin, Denis Kuznedelev, Konstantin Burlachenko, Kai Yi, Dan Alistarh, and Peter Richtárik. PV-Tuning: Beyond straight-through estimation for extreme LLM compression. *ArXiv*, abs/2405.14852, 2024.

[28] Raphael A Meyer, Cameron Musco, Christopher Musco, and David P Woodruff. Hutch++: Optimal stochastic trace estimation. In *Proceedings of the 2021 ACM-SIAM Symposium on Simplicity in Algorithms (SOSA)*, pages 142–155. SIAM, 2021. doi: 10.1137/1.9781611976496.16.

[29] Jorn Peters, Marios Fournarakis, Markus Nagel, Mart van Baalen, and Tijmen Blankevoort. QBitOpt: Fast and accurate bitwidth reallocation during training. *arXiv:2307.04535*, 2023.

[30] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

[31] Soroush Tabesh, Mher Safaryan, and Dan Alistarh. CAGE: Curvature-aware gradient estimation for accurate quantization-aware training, 2025. URL `https://arxiv.org/abs/2510.18784`.

[32] Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP#: Even better LLM quantization with Hadamard incoherence and lattice codebooks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 48630–48656, 2024.

[33] Feiyu Wang, Xinyu Tan, Bokai Huang, Yihao Zhang, Guoan Wang, Peizhuang Cong, and Tong Yang. Fairy2i: Training complex LLMs from real LLMs with all parameters in $\{\pm 1, \pm i\}$, 2025. URL `https://arxiv.org/abs/2512.02901`.

[34] Feiyu Wang, Guoan Wang, Yihao Zhang, Shengfan Wang, Weitao Li, Bokai Huang, Shimao Chen, Zihan Jiang, Rui Xu, and Tong Yang. iFairy: The first 2-bit complex LLM with all parameters in $\{\pm 1, \pm i\}$, 2025. URL `https://arxiv.org/abs/2508.05571`.

[35] Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. BitNet: Scaling 1-bit transformers for large language models. *arXiv preprint arXiv:2310.11453*, 2023.

[36] Ruizhe Wang, Yeyun Gong, Peng Cheng, et al. Optimizing large language model training using FP4 quantization. *arXiv preprint arXiv:2501.17116*, 2025. Microsoft Research.

[37] Yudong Wang, Zixuan Fu, Jie Cai, Peijun Tang, Hongya Lyu, Yewei Fang, Zhi Zheng, Jie Zhou, Guoyang Zeng, Chaojun Xiao, et al. Ultra-FineWeb: Efficient data filtering and verification for high-quality LLM training data. *arXiv preprint arXiv:2505.05427*, 2025.

[38] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

A Implementation Details

We provide the hyperparameter configuration used for HESTIA training in Table 8. The base initial temperature τ_{init} is fixed to 0.3 across all main experiments and held constant during the compress stage, while the tensor-wise schedules are determined by the offline Hutch++ sensitivity scores and the scaling strength α in Eq. (15).

Table 8: Hyperparameter configuration for HESTIA quantization-aware training.

General Training Settings		*Method-Specific Settings*	
Hyperparameter	Value	Hyperparameter	Value
Global Batch Size	256	Pressure Ratio ρ	0.2
Optimizer	AdamW	Temperature-Scaling Strength α	0.4
Learning Rate	5e-5	Base Initial Temperature	0.3
LR Scheduler	WSD	Temperature Schedule	Cosine
Weight Decay	0.01	Group Size	128
Sensitivity Estimation (Hutch++)			
Calibration Batch Size	50	Sketch Rank r	10
Hutchinson Samples	20	Noise Type	Rademacher
Sensitivity Gain κ	1.0	Normalization Epsilon ϵ_h	1e-8

B Proofs for Sec. 4

B.1 Proof of Lemma 4.1

Proof. Recall $\pi_\tau(q \mid w)$ in Eq. (7) and $\mathcal{H}(w;\tau)$ in Eq. (8). Throughout, we treat γ as a constant during differentiation as in Sec. 3.1. We prove that

$$\frac{\partial \mathcal{H}(w;\tau)}{\partial w} = \frac{2}{\tau}\,\mathbb{V}_\tau(w),$$

where $\mu_\tau(w)$ and $\mathbb{V}_\tau(w)$ are defined in Eqs. (16)–(17).

Step 1: Reparameterization. Let $z \triangleq w/\gamma$. Since γ is constant,

$$\frac{\partial z}{\partial w} = \frac{1}{\gamma}. \tag{23}$$

Define $a_q(z) \triangleq \exp\!\left(-\frac{(z-q)^2}{\tau}\right)$ and $Z(z) \triangleq \sum_{k\in\mathcal{Q}} a_k(z)$. Then $\pi_\tau(q \mid w) = \pi_\tau(q \mid z) = a_q(z)/Z(z)$.

Step 2: Derivative of $\pi_\tau(q \mid z)$. Differentiate $a_q(z)$:

$$\frac{\partial a_q(z)}{\partial z} = -\frac{2(z-q)}{\tau}\,a_q(z). \tag{24}$$

Hence

$$\frac{\partial Z(z)}{\partial z} = \sum_{k\in\mathcal{Q}} \frac{\partial a_k(z)}{\partial z} = -\frac{2}{\tau}\sum_{k\in\mathcal{Q}} (z-k)\,a_k(z). \tag{25}$$

Applying the quotient rule to $\pi_\tau(q \mid z) = a_q(z)/Z(z)$ gives

$$\begin{aligned}\frac{\partial}{\partial z}\pi_\tau(q \mid z) &= \frac{(\partial_z a_q)Z - a_q(\partial_z Z)}{Z^2} \\ &= -\frac{2}{\tau}\pi_\tau(q \mid z)(z-q) + \frac{2}{\tau}\pi_\tau(q \mid z)\sum_{k\in\mathcal{Q}}(z-k)\pi_\tau(k \mid z).\end{aligned} \tag{26}$$

Since $\sum_k \pi_\tau(k \mid z) = 1$ and $\mu_\tau(w) = \sum_k k\pi_\tau(k \mid w) = \sum_k k\pi_\tau(k \mid z)$,

$$\sum_{k\in\mathcal{Q}} (z-k)\pi_\tau(k \mid z) = z - \mu_\tau(w).$$

Substituting into Eq. (26) yields

$$\frac{\partial}{\partial z}\pi_\tau(q \mid z) = \frac{2}{\tau}\pi_\tau(q \mid z)\big(q - \mu_\tau(w)\big). \tag{27}$$

Step 3: Derivative of $\mu_\tau(w)$. By Eq. (16), $\mu_\tau(w) = \sum_{q\in\mathcal{Q}} q\,\pi_\tau(q \mid z)$, hence

$$\begin{aligned}
\frac{\partial \mu_\tau(w)}{\partial z} &= \sum_{q\in\mathcal{Q}} q\,\frac{\partial}{\partial z}\pi_\tau(q \mid z) \\
&= \sum_{q} q \cdot \frac{2}{\tau}\pi_\tau(q \mid z)\big(q - \mu_\tau(w)\big) \\
&= \frac{2}{\tau}\left(\sum_q q^2\pi_\tau(q \mid z) - \mu_\tau(w)\sum_q q\pi_\tau(q \mid z)\right) \\
&= \frac{2}{\tau}\left(\sum_q q^2\pi_\tau(q \mid w) - \mu_\tau(w)^2\right) = \frac{2}{\tau}\,\mathbb{V}_\tau(w),
\end{aligned} \tag{28}$$

where the last equality uses Eq. (17).

Step 4: Conclude $\partial_w\mathcal{H}(w;\tau)$. Since $\mathcal{H}(w;\tau) = \gamma\,\mu_\tau(w)$ in Eq. (8) and γ is constant,

$$\begin{aligned}
\frac{\partial \mathcal{H}(w;\tau)}{\partial w} &= \gamma\frac{\partial \mu_\tau(w)}{\partial w} = \gamma\frac{\partial \mu_\tau(w)}{\partial z}\cdot\frac{\partial z}{\partial w} \\
&= \gamma\cdot\frac{2}{\tau}\mathbb{V}_\tau(w)\cdot\frac{1}{\gamma} = \frac{2}{\tau}\,\mathbb{V}_\tau(w),
\end{aligned} \tag{29}$$

where we used Eq. (23) and Eq. (28). This proves Lemma 4.1. □

B.2 Proof of Lemma 4.2

Proof. Recall $\mathcal{H}(w;\tau)$ in Eq. (8) and the hard quantizer $Q(w)$ in Eq. (2), with boundaries $\mathcal{B} = \{-\gamma/2, +\gamma/2\}$ in Eq. (5). We prove that, as $\tau \to 0^+$,

$$\frac{\partial \mathcal{H}(w;\tau)}{\partial w} \xrightarrow{\mathcal{D}'} \gamma\sum_{b\in\mathcal{B}}\delta(w-b).$$

Step 1: Piecewise form of $Q(w)$. From Eq. (2), for scalar w let $z = w/\gamma$. Then $Q(w) = \gamma\cdot\texttt{Clip}(\texttt{Round}(z), -1, +1)$, which implies

$$Q(w) = \begin{cases} -\gamma, & w < -\gamma/2, \\ 0, & -\gamma/2 < w < \gamma/2, \\ +\gamma, & w > \gamma/2. \end{cases} \tag{30}$$

Any additional rounding transitions (e.g., at $|z| = 1.5$) do not change the clipped output and hence do not introduce discontinuities.

Step 2: Pointwise limit $\mathcal{H}(w;\tau) \to Q(w)$ for $w \notin \mathcal{B}$. Let $z = w/\gamma$ and define $d_q(z) \triangleq (z-q)^2$ for $q \in \mathcal{Q}$. If $z \notin \{-\frac{1}{2}, +\frac{1}{2}\}$ (equivalently $w \notin \mathcal{B}$), the minimizer $q^\star(z) \in \arg\min_{q\in\mathcal{Q}} d_q(z)$ is unique. For any $q \neq q^\star(z)$ define $\Delta_q(z) \triangleq d_q(z) - d_{q^\star(z)}(z) > 0$. Using Eq. (7),

$$\frac{\pi_\tau(q \mid w)}{\pi_\tau(q^\star(z) \mid w)} = \exp\!\left(-\frac{\Delta_q(z)}{\tau}\right) \xrightarrow[\tau\to 0^+]{} 0. \tag{31}$$

Since $\sum_{q\in\mathcal{Q}}\pi_\tau(q \mid w) = 1$, Eq. (31) implies $\pi_\tau(q^\star(z) \mid w) \to 1$ and $\pi_\tau(q \mid w) \to 0$ for all $q \neq q^\star(z)$. Therefore, for all $w \notin \mathcal{B}$,

$$\lim_{\tau\to 0^+}\mathcal{H}(w;\tau) = \gamma\sum_{q\in\mathcal{Q}} q\lim_{\tau\to 0^+}\pi_\tau(q \mid w) = \gamma\,q^\star(w/\gamma). \tag{32}$$

Comparing Eq. (32) with Eq. (30) yields $\lim_{\tau\to 0^+} \mathcal{H}(w;\tau) = Q(w)$ for all $w \neq \pm\gamma/2$, hence the convergence holds almost everywhere:

$$\mathcal{H}(w;\tau) \xrightarrow[\tau\to 0^+]{} Q(w) \quad \text{for a.e. } w \in \mathbb{R}. \tag{33}$$

Step 3: Distributional convergence of derivatives. Let $\varphi \in C_c^\infty(\mathbb{R})$ be any test function. For each $\tau > 0$, $\mathcal{H}(\cdot;\tau)$ is smooth, thus

$$\left\langle \frac{\partial \mathcal{H}(\cdot;\tau)}{\partial w}, \varphi \right\rangle = -\left\langle \mathcal{H}(\cdot;\tau), \varphi' \right\rangle = -\int_{\mathbb{R}} \mathcal{H}(w;\tau)\, \varphi'(w)\, dw. \tag{34}$$

Since $|\mathcal{H}(w;\tau)| \leq \gamma$ and φ' has compact support, $|\mathcal{H}(w;\tau)\varphi'(w)| \leq \gamma|\varphi'(w)|$ with $\gamma|\varphi'| \in L^1(\mathbb{R})$. Together with the a.e. convergence in Eq. (33), the dominated convergence theorem yields

$$\lim_{\tau\to 0^+} \int_{\mathbb{R}} \mathcal{H}(w;\tau)\, \varphi'(w)\, dw = \int_{\mathbb{R}} Q(w)\, \varphi'(w)\, dw. \tag{35}$$

Combining Eqs. (34)–(35),

$$\lim_{\tau\to 0^+} \left\langle \frac{\partial \mathcal{H}(\cdot;\tau)}{\partial w}, \varphi \right\rangle = -\int_{\mathbb{R}} Q(w)\, \varphi'(w)\, dw. \tag{36}$$

Step 4: Evaluate the limit functional. Using the piecewise form in Eq. (30),

$$\begin{aligned} \int_{\mathbb{R}} Q(w)\, \varphi'(w)\, dw &= \int_{-\infty}^{-\gamma/2} (-\gamma)\, \varphi'(w)\, dw + \int_{\gamma/2}^{\infty} (+\gamma)\, \varphi'(w)\, dw \\ &= -\gamma\big(\varphi(-\gamma/2) - \varphi(-\infty)\big) + \gamma\big(\varphi(\infty) - \varphi(\gamma/2)\big). \end{aligned} \tag{37}$$

Because φ has compact support, $\varphi(\pm\infty) = 0$, hence

$$\int_{\mathbb{R}} Q(w)\, \varphi'(w)\, dw = -\gamma\, \varphi\Big(-\frac{\gamma}{2}\Big) - \gamma\, \varphi\Big(+\frac{\gamma}{2}\Big). \tag{38}$$

Plugging Eq. (38) into Eq. (36) yields

$$\lim_{\tau\to 0^+} \left\langle \frac{\partial \mathcal{H}(\cdot;\tau)}{\partial w}, \varphi \right\rangle = \gamma\, \varphi\Big(-\frac{\gamma}{2}\Big) + \gamma\, \varphi\Big(+\frac{\gamma}{2}\Big) = \left\langle \gamma\delta\Big(w + \frac{\gamma}{2}\Big) + \gamma\delta\Big(w - \frac{\gamma}{2}\Big),\, \varphi(w) \right\rangle.$$

Since φ is arbitrary, we conclude

$$\frac{\partial \mathcal{H}(w;\tau)}{\partial w} \xrightarrow[\tau\to 0^+]{} \gamma \sum_{b\in\mathcal{B}} \delta(w - b),$$

which is Eq. (20) and completes the proof of Lemma 4.2. □

B.3 Proof of Proposition 4.3

Proof. For $q \neq q^\star$, the probability ratio is $\pi_\tau(q \mid w)/\pi_\tau(q^\star \mid w) = e^{-\Delta_q/\tau}$, where $\Delta_q \geq \Delta$. Since $\pi_\tau(q^\star \mid w) \leq 1$, the total mass outside $q^\star$ is at most $(K-1)e^{-\Delta/\tau}$. Multiplying this mass by $\gamma \max_q |q - q^\star| \leq \gamma\, \mathrm{diam}(\mathcal{Q})$ bounds the deviation of the expectation from $\gamma q^\star$. □

C Hessian-Guided Scheduling Analysis

This section provides additional evidence for the Hessian-guided annealing design in Sec. 3.3. Hessian trace provides a tensor-wise sensitivity prior for scheduling the soft-to-hard transition. We therefore examine two questions: whether LLM tensors exhibit sufficiently heterogeneous sensitivity, and whether Hessian trace provides a useful scheduler signal compared with simpler statistics.

Figure 4 shows that curvature sensitivity is highly non-uniform across projection types and layers. This heterogeneity motivates testing a tensor-wise schedule: the same annealing rate may be too aggressive for high-trace tensors and unnecessarily slow for low-trace tensors. HESTIA uses this

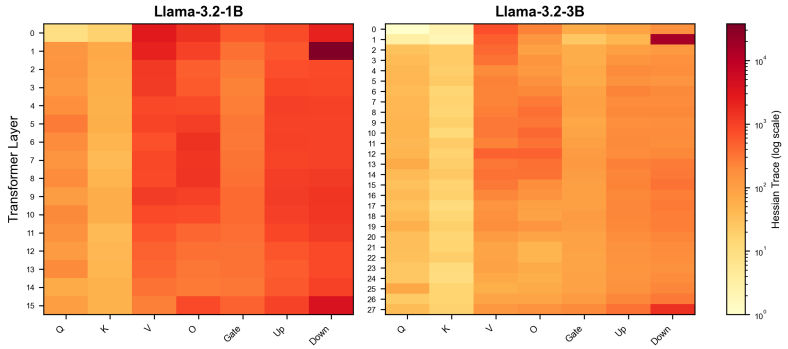


Figure 4: Visualization of tensor-wise Hessian traces across linear projections in Llama-3.2. V and Down projections consistently exhibit the largest trace values, and substantial variation exists both across layers and within layers. This motivates tensor-wise annealing in HESTIA: sensitive tensors can remain in the smooth regime longer, while less sensitive tensors can harden earlier.

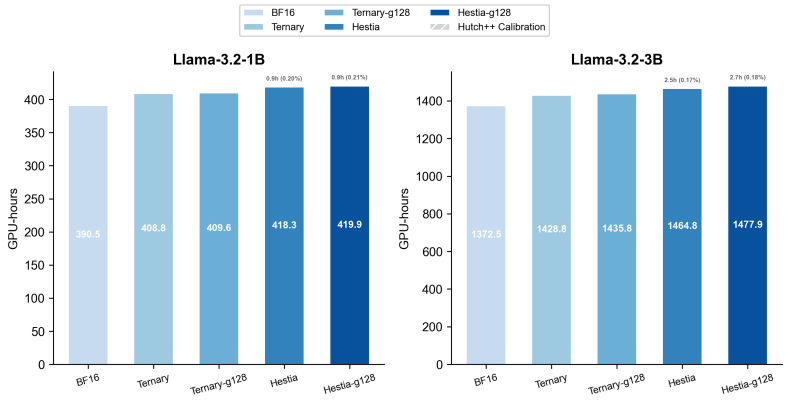


Figure 5: Training cost and the calibration component, in GPU-hours. The approximately 0.2% labels refer only to Hutch++ calibration. Relative to the same grouped-ternary baseline, total HESTIA-g128 overhead is approximately 2.5% at 1B and 2.9% at 3B.

structure only to assign annealing rates, leaving the quantization codebook and inference operator unchanged.

Figure 3 compares the Hessian trace with gradient norm and activation variance. The observed Spearman rank correlation between Hessian trace and gradient norm is 0.973. Hessian trace exhibits a wider dynamic range in the calibration distribution. The downstream comparison in Table 1 gives 0.543 for GradNorm and 0.547 for Hessian scheduling. We use Hessian trace as the default, while GradNorm offers a cheaper alternative. Both variants use scores computed once before QAT.

D Computational Efficiency Results

This section complements the efficiency discussion in Sec. 5.7. HESTIA adds two sources of work relative to standard QAT: a one-time offline sensitivity calibration stage and lightweight element-wise relaxed-quantizer operations during training. Calibration is performed once before QAT, and the resulting tensor-wise sensitivity scores are fixed throughout training.

All training and calibration costs are measured in GPU-hours on NVIDIA RTX 4090 GPUs with 24GB memory. Figure 5 reports 409.6/419.9 GPU-hours for ternary-g128/HESTIA-g128 at 1B, and 1435.8/1477.9 at 3B. These give relative overheads of 2.51% and 2.93%. Calibration contributes 0.9 and 2.7 GPU-hours (approximately 0.2%), with the remaining overhead arising during relaxed-quantizer training. The stored sensitivity map contains one scalar score per calibrated tensor. Inference uses the selected hard format and adds no continuation-specific operation; efficient execution requires a suitable grouped-ternary implementation. The reported timings cover the main training and calibration comparison.

Table 9: Literature context: zero-shot accuracy from heterogeneous training protocols. #Tokens denotes each report's nominal budget. Model sizes, data, and optimization follow the respective studies. HESTIA rows are five-run means; other rows follow prior reports.

Method	Size	#Tokens	ARC-E	ARC-C	HellaSwag	PIQA	WinoGrande	Avg. (↑)
BF16	1B	–	**0.654**	**0.313**	**0.477**	**0.743**	**0.603**	**0.558**
LSQ	1B	10B	0.376	0.177	0.258	0.574	0.506	0.378
SEQ (ParetoQ)	1B	10B	0.421	0.180	0.273	0.604	0.510	0.398
DLT	1B	10B	0.424	0.174	0.256	0.563	0.513	0.386
TWN	1B	10B	0.407	0.220	0.284	0.601	0.492	0.401
AbsMean	1B	10B	0.603	0.259	0.360	0.683	0.541	0.489
LLM-QAT	1B	100B	0.360	0.262	0.313	0.551	0.496	0.397
BitNet	1.3B	100B	0.549	0.242	0.377	0.688	**0.558**	0.483
Spectra	1.1B	100B	0.563	0.246	0.388	0.693	0.555	0.489
Tequila	1B	10B	0.645	0.305	**0.391**	**0.710**	0.542	**0.519**
Sherry	1B	10B	**0.647**	**0.309**	0.388	0.699	0.550	**0.519**
HESTIA	1B	10B	**0.665**	**0.350**	**0.418**	**0.726**	**0.577**	**0.547**
BF16	3B	–	**0.745**	**0.422**	**0.552**	**0.768**	**0.691**	**0.636**
LSQ	3B	10B	0.431	0.200	0.294	0.599	0.522	0.409
SEQ (ParetoQ)	3B	10B	0.498	0.231	0.303	0.645	0.529	0.441
DLT	3B	10B	0.361	0.161	0.260	0.572	0.496	0.370
TWN	3B	10B	0.692	0.351	0.462	0.734	0.586	0.565
AbsMean	3B	10B	0.672	0.329	0.439	0.735	0.582	0.551
LLM-QAT	3B	100B	0.445	0.307	0.434	0.627	0.506	0.464
BitNet	3B	100B	0.614	0.283	0.429	0.715	0.593	0.527
Spectra	3.9B	100B	0.660	0.319	**0.483**	**0.744**	**0.631**	0.567
Tequila	3B	10B	**0.702**	0.346	0.464	0.739	**0.627**	**0.576**
Sherry	3B	10B	0.688	**0.364**	0.452	0.736	0.593	0.567
HESTIA	3B	10B	**0.741**	**0.393**	**0.491**	**0.757**	0.622	**0.601**

E Additional Literature Comparisons

Table 9 presents the cross-paper comparison for context. Baselines include LSQ [11], Stretched Elastic Quantization (SEQ) as used in ParetoQ [23], DLT [5], TWN [21], LLM-QAT [22], BitNet [26], Spectra [18], Tequila [15], and Sherry [16]. Most comparator values follow the compilation in Tequila; Sherry follows its own report. The published Tequila rows (0.519/0.576) differ from the controlled 1B rerun (0.530) in Table 1. Spectra also uses different model sizes and 100B tokens. Table 1 provides the matched-training comparison. BF16 serves as the full-precision reference.

F Relation to Prior Quantization Components

Table 10 separates the components of the continuation recipe. The comparison describes the representative mechanisms in each cited formulation. In particular, for a fixed scalar codebook, Eq. (7) shares the assignment primitive of Agustsson et al.; Hessian-guided control is also part of a broader history of sensitivity-based quantization.

G Response and Boundary Diagnostics

Population measurement. Table 6 covers 973,078,528 scalar weights in 112 projection tensors of the examined 1B checkpoints. Each uses its own group scale to form $z = w/\gamma$; the normalized ternary boundaries are ± 0.5. The fractions and mean margins describe the full selected population. The step-31,000 check (23.56% STE, 3.67% HESTIA within distance 0.1) supplies an additional preterminal comparison.

Matched-update protocol. The 100-step diagnostic uses a schedule rescaled to $T = 100$ (approximately 26.2M token slots). It fixes 512 weight indices in each of 21 tensors and observes windows 5–14, 45–54, and 85–94, giving 210 layer-step clusters per window. The primary measurements apply each STE-trajectory latent update to both the hard map and the HESTIA continuation map. Scale, pressure, and temperature are frozen between the pre-update and post-update evaluations, so schedule or scale changes alone are not counted as update response. We retain $|\Delta w| > 10^{-7}$ and call

Table 10: Component-level context. "Pressure" denotes interpolation of latent and quantized/relaxed weights. Entries describe the cited formulations.

Method	Assignment/forward map	Pressure	Hardening control	Sensitivity role	Endpoint
Agustsson et al. [1]	Distance Softmax expectation	No separate pressure term	Soft-to-hard annealing	No Hessian temperature prior	Learned centers, hard assignment
Louizos et al. [25]	Probabilistic relaxed discrete weights	No HESTIA pressure term	Relaxation temperature	No Hessian temperature prior	Discrete quantized weights
Liu–Mattina [24]	FP/hard-quantized blend	Alpha-blending	Increasing blend coefficient	No Hessian temperature prior	Hard quantized weights
DSQ [13]	Interval-wise smooth map; hard forward	No HESTIA pressure term	Softness of approximation	No Hessian temperature prior	Hard quantized weights
HAWQ / QBitOpt [8, 29]	Quantizers at allocated bit-widths	Not their primary control	Precision allocation	Curvature or sensitivity controls bit-width	Allocated discrete formats
Bin Regularization [14]	Hard-STE plus bin penalty	No HESTIA pressure term	Auxiliary regularization	Bin-based penalty	Hard quantized weights
HESTIA	Fixed-codebook Softmax expectation	Linear p_t	Cosine $\bar{\tau}(t)$	Offline trace modulates τ_i	Prescribed hard codebook/scale rule

a change a response only if its absolute magnitude exceeds 10^{-12}. Both maps are evaluated on the same STE-trajectory updates.

Table 7 in the main text reports the retained update counts, no-response events, and 95% cluster-bootstrap intervals for each window.

A separate HESTIA trajectory gives response fractions of 100%, 77.88%, and 13.32% in the three windows. These measurements use the HESTIA trajectory's own updates. An update-filter sensitivity check used $0, 10^{-8}, 10^{-7}, 10^{-6}$; the response threshold remained 10^{-12}. These diagnostics characterize the 100-step schedule.

Terminal continuation. A separate trajectory uses 8.307B tokens with $T = 31{,}690$. Restoring the model, optimizer, scheduler, and random-number-generator state at step 31,000 allows 690 updates through the endpoint. For $|\Delta w| > 10^{-7}$, the recorded response fractions are 0% for the hard map and 0.00044% for HESTIA. This near-discrete terminal behavior complements the early/middle-window response measured in the short diagnostic. The 8.307B trajectory is reported separately from the nominal 10B main protocol; at batch 256 and sequence length 1024, $31{,}690 \times 256 \times 1024 = 8{,}307{,}343{,}360$ token slots.

Jacobian localization. On the step-31,000 checkpoint, the distance-binned Jacobian magnitudes decrease with distance from a boundary. At $\tau = 0.01$, every measured bin beyond normalized distance 0.1 has mean absolute Jacobian below 6×10^{-9}. This is consistent with the localization result. The theory characterizes the operator derivative; checkpoint distributions additionally depend on the task loss and optimizer.

H Extensions and Transfer Limits

H.1 Nonuniform Codebooks and Hard Endpoints

Section 5.6 describes the nonuniform real-codebook construction and its hard endpoint. Logarithmically spaced levels can be inserted directly into this squared-distance construction. Applying a logarithmic transform to the distance instead would change the metric, boundaries, and derivative through a chain-rule factor; it is a different quantizer unless the target also uses that metric. For a complex codebook, assignments can use squared Euclidean distances in the real-imaginary plane, with the scalar variance identity replaced by a covariance Jacobian.

NF4 proof of concept. We use the 16 NormalFloat4 levels introduced with QLoRA [7], group size 64, absmax scaling, and nearest-code lookup at the endpoint. An endpoint check against bitsandbytes over 1,048,576 values is bit-exact, with maximum absolute error 0. The nonuniform spacing changes local assignment competition but not the fixed-codebook construction.

A 2,000-step Qwen2.5-0.5B study uses 512,000 non-repeated packed sequences (approximately 524M token slots) and a disjoint 256-sequence validation set. All losses and gradient norms remain finite. Final scheduled and forced-hard validation NLL are both 3.26766 (perplexity 26.25); the endpoint NLL gap and weight mean-squared error are both zero. This study demonstrates compatibility with the NF4 codebook and convergence to its hard endpoint.

H.2 Untuned MoE Transfer

We study the public `Granite-3.1-1B-A400M-Base` checkpoint [17], which has approximately 1.33B total parameters, 24 layers, 32 experts per layer, and top-8 routing. On 16,384 real-text tokens without weight updates, the median expert max/min token-load ratio is $74.9\times$ (95th percentile $297.6\times$), while the median router-gradient max/min ratio is $68.2\times$. Their rank correlation is 0.65. This uneven support limits how confidently a one-time calibration batch can characterize individual experts.

A matched 200-step continuation implements the fused three-dimensional expert tensors while retaining routers and attention in BF16. Table 11 shows that HESTIA avoids the immediate degradation of hard STE and finishes with lower NLL than STE, but remains substantially worse than BF16. All runs have finite losses, and HESTIA's final forced-hard NLL gap is 0.0077. The runs use the dense-model schedule without MoE-specific tuning, and their initial effective weight maps differ. Load-aware calibration, estimator-confidence normalization, routing-aware schedules, and dynamic refresh remain possible extensions.

Table 11: Untuned 200-step MoE continuation. BF16 provides the full-precision reference.

Expert weight path	Initial validation NLL	Final validation NLL
BF16	2.309	2.364
HESTIA	2.309	5.451
Hard STE	10.518	6.081

H.3 Schedule Heuristics and Activation Quantization

The dense-model defaults $(\rho, \alpha, \tau_{\text{init}}) = (0.2, 0.4, 0.3)$ are selected from the local sweeps in Table 5. A useful interpretation of α is to choose a maximum temperature ratio $r_{\max}$ and set $\alpha = \log r_{\max}$; $r_{\max} = 1.5$ gives about 0.405. For adjacent code spacing δ, at a code center the neighboring assignment's unnormalized weight ratio is $r = e^{-\delta^2/\tau}$, suggesting $\tau_{\text{init}} = \delta^2/\log(1/r)$. With ternary spacing 1 and temperature 0.3, $r \approx 0.036$. Nonuniform codebooks have several spacings and cannot generally realize the same ratio everywhere. These heuristics provide starting points for schedule selection; evaluating them at 70B scale remains future work.

The same expectation can be applied to activations with a prescribed activation scale, but joint weight–activation QAT changes the calibration problem. Input-dependent ranges, outliers, scale gradients, and compounded weight/activation noise would need separate treatment. Our experiments use weight-only quantization; W1.58A1.58 is a potential extension.

I Reporting Conventions and Code Availability

The main HESTIA rows report means over five training runs. The additional controlled baselines and ablations report point estimates. The intervals in Appendix G are obtained by bootstrapping layer-step clusters from the short operator-response diagnostic. Boundary fractions summarize the examined weight population. The 100-step diagnostic, the 8.307B-token terminal trajectory, and the NF4/MoE proofs of concept have their own stated budgets and are distinct from the nominal 10B-token main comparison.

The public repository linked in the abstract provides an example implementation. The continuation equations in Sec. 3.2, tensor-wise schedule in Sec. 3.3, and hyperparameters in Appendix A specify the method used in the paper. These include detached group scales, $\epsilon_\gamma = 10^{-8}$, and a tensor-wise temperature multiplier throughout annealing. The public code demonstrates one example case of this method.

J Broader Impact

Lower-bit weights may reduce the storage and execution cost of language models when suitable kernels are available, which can broaden access to local inference. Quantization does not remove model biases, factual errors, privacy risks, or potential misuse. Accuracy changes can also vary across tasks and user groups, so aggregate benchmark improvements should not substitute for deployment-specific quality and safety evaluations. The extra QAT compute must be considered alongside any inference savings; this study does not measure net energy savings or environmental impact.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction distinguish the continuation recipe from prior Softmax assignments, empirical boundary separation from theoretical guarantees, and point estimates from five-run means.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: Sec. 6 and Appendices H–I discuss static calibration, model scale, weight-only quantization, transfer limits, and the scope of the example code.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes].

   Justification: Sec. 4 states fixed positive scale and finite-temperature assumptions, the tie exception, and the scope of the boundary limit; Appendix B proves the Jacobian, localization, and soft-to-hard error bound.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [No].

   Justification: The continuation method and principal hyperparameters are specified, but we do not provide complete run-specific settings for every added baseline and extension; Appendix I describes the reporting and code scope.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No].

   Justification: The public repository provides an example implementation rather than executable configurations for all reported experiments; Appendix I states its scope.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [No].

   Justification: Sec. 5.1 and Appendix A give the principal settings. Complete details for all added comparisons, including the exact tuned DSQ-style parameters, are not reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No].

   Justification: Main results are reported as five-run means or point estimates. Table 7 includes 95% cluster-bootstrap intervals for the operator-response diagnostic; training-seed intervals are not reported.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No].

   Justification: Appendix D reports RTX 4090 (24GB) GPU-hours for the main training and calibration comparison, but does not report complete resource accounting for all added studies, tuning, and the full project.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The work uses public model checkpoints and benchmark datasets for standard quantization and evaluation research, and does not involve deceptive practices, unsafe deployment, or human-subject experimentation.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: Appendix J discusses possible access and storage benefits, inherited model risks, task-dependent degradation, deployment evaluation, and the absence of a net-energy assessment.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper releases a quantization training method rather than a high-risk pretrained model, image generator, scraped dataset, or user-facing system requiring dedicated misuse safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No].

    Justification: The paper cites the pretrained checkpoints, training data, evaluation harness, and benchmark datasets used in the experiments, but it does not explicitly enumerate the license terms for each existing asset.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No].

    Justification: The method and example code are documented, but the public release does not include configurations and documentation for every experiment; Appendix I describes this scope.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The work does not involve crowdsourcing, participant studies, or other research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The work does not involve crowdsourcing, participant studies, or other research with human subjects, so IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: LLMs are the target models for quantization and evaluation, but they are not used as an important, original, or non-standard component in the core method development.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.