BEYOND EXACT EQUIVALENCE: RETHINKING QUANTIZATION PREPROCESSING

Anonymous authors
Paper under double-blind review

ABSTRACT

Most quantization preprocessing methods search for transformations that exactly preserve the pretrained model's full-precision function. However, the model that matters is the one after quantization: why should optimization remain confined to equivalence before rounding? We challenge this restriction and formulate quantization preprocessing as optimization over a broader space of representations, where exact equivalence is only one possible constraint. Our framework accommodates flexible offline preprocessing, nonlinear parameter transformations, and direct weight adaptation while keeping deployment constraints explicit. We instantiate this perspective with an equivalence-preserving structured initialization followed by global quantization-aware distillation, allowing the full-precision parameters to deviate from the original model to better preserve its behavior after quantization. Across multiple model families, this relaxation consistently enhances post-quantization language-modeling quality compared to matched post-training quantization baselines, without requiring preprocessing-specific online transformations at deployment. These results establish a broader principle for low-bit compression: exact equivalence can provide a safe starting point, but should not define the space in which the quantized model is optimized.

1 INTRODUCTION

Low-bit quantization reduces the storage and deployment costs of large language models by representing weights and activations with fewer bits. Its central challenge is to preserve the model's behavior despite the loss of numerical precision. This is particularly difficult for low-bit weight-and-activation quantization, where a small number of large-magnitude outliers can dominate the quantization range and reduce resolution for the remaining values (Xiao et al., 2023; Ashkboos et al., 2024). A major line of work, therefore, preprocesses pretrained models into representations that are easier to quantize.

Most existing preprocessing methods operate under a common constraint: they change the representation while preserving the full-precision function exactly. SmoothQuant redistributes the scale between activations and weights, QuaRot applies fixed rotations, and SpinQuant learns rotations using a quantized-network objective (Xiao et al., 2023; Ashkboos et al., 2024; Liu et al., 2025). These methods exploit the fact that functionally equivalent full-precision representations can behave differently after rounding. Exact equivalence thus provides a safe way to improve quantization without changing the original full-precision model.

But exact equivalence before rounding and fidelity after rounding are different requirements. The deployed model is the quantized model, and a representation that slightly departs from the original full-precision function may better preserve its behavior after quantization. Enforcing exact equivalence throughout the optimization can, therefore, exclude useful solutions and restrict preprocessing to transformation families that admit exact compensation. This motivates our central question:

> *Is exact preservation of the full-precision function necessary for obtaining the best post-quantization model?*

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

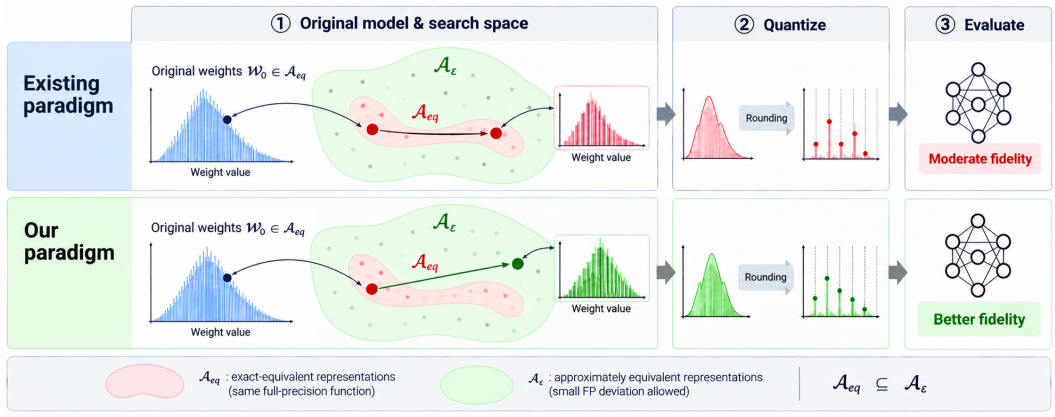


Figure 1: Comparison of quantization preprocessing paradigms. (Top) Existing paradigm restricts optimization within functionally exact-equivalent representations ($\mathcal{A}_{eq}$), leading to a residual fidelity gap after low-bit rounding. (Bottom) Our proposed paradigm use exact equivalence solely as an initialization ($\mathcal{W}_0 \in \mathcal{A}_{eq}$) and relaxes the optimization space to approximately equivalent representations ($\mathcal{A}_\epsilon$) or even unrestricted ones ($\mathcal{A}_\infty$), achieving significantly better post-quantization fidelity under identical deployment constraints.

Relaxing this constraint could open a broader design space, including nonlinear offline parameter transformations and direct weight adaptation, while preserving the deployment constraints that actually matter.

We formalize this idea by separating *full-precision equivalence* from *post-quantization fidelity*. Our formulation treats equivalence-preserving transformations as a restricted family within a broader optimization space that permits auxiliary preprocessing, nonlinear offline parameter maps, joint optimization, and direct weight adaptation under explicit quantization and deployment constraints. This flexibility need not introduce additional inference-time computation because transformations used only to construct the final parameters can remain entirely offline. The key distinction is therefore not whether the objective accounts for quantization, SpinQuant already optimizes a quantized-network objective (Liu et al., 2025), but *which full-precision representations the optimization is allowed to explore*. Related quantization-aware training and post-rotation adaptation methods provide complementary precedents (Liu et al., 2024; Tseng et al., 2024; Chen et al., 2025).

We conceptually compare the search space of the existing paradigm and our proposed relaxed optimization paradigm in Figure 1. We study a simple realization of this principle: an exactly equivalent structured initialization followed by global quantization-aware distillation. After initialization, the weights are free to depart from the original full-precision function and are optimized to better match the teacher *after quantization*. The resulting deployment graph requires no preprocessing-specific online rotations. Across multiple model families, this relaxation consistently improves post-quantization language-modeling quality over matched post-training baselines, providing empirical evidence that strong quantized solutions need not remain within the exactly equivalent representation family.

Our contributions are threefold:

- **We challenge exact full-precision equivalence as a necessary constraint for quantization**, motivating optimization directly for post-quantization fidelity.
- **We introduce a general formulation beyond exact equivalence**, enabling more flexible offline preprocessing and weight adaptation under fixed deployment constraints.
- **We show that relaxing equivalence improves quantized models in practice**, with consistent gains in post-quantization language-modeling quality across multiple model families.

Taken together, our results suggest a broader principle for quantization preprocessing: exact equivalence can provide a safe starting point but need not define the space in which the final quantized model is optimized.

2 BACKGROUND: WHAT PREPROCESSING CHANGES

2.1 QUANTIZATION CHANGES THE NUMERICAL REPRESENTATION

Full precision (FP) denotes the unquantized reference computation. A *weight* is a stored model parameter, whereas an *activation* is an intermediate value produced during a forward pass. Quantization replaces selected weights and activations with values from a discrete set. For a positive integer bit width b and real clipping endpoints $\ell < h$, an illustrative uniform grid is

$$\Delta = \frac{h-\ell}{2^b-1}, \qquad \{\ell + k\Delta : k = 0, \dots, 2^b - 1\}. \tag{1}$$

Values are clipped to $[\ell, h]$ and rounded to the nearest grid point under a fixed tie-breaking rule. Practical uniform quantizers usually specify a scale and, for asymmetric quantization, a zero point; a *quantization group* is a set of entries sharing these parameters (Jacob et al., 2018). Appendix B.2 gives the scale/zero-point formulation.

At a fixed bit width, a wider clipping interval produces a coarser grid. A few large-magnitude values can therefore reduce the resolution available to the remaining values. Clipping more aggressively trades this loss of resolution for clipping error (Dettmers et al., 2022; Xiao et al., 2023). Preprocessing addresses this difficulty by changing the representation of weights and activations before quantization; it need not change the target bit width or the quantization rule.

2.2 EQUIVALENT COMPUTATION NEED NOT IMPLY EQUIVALENT QUANTIZATION

Consider a linear projection xW, where $x \in \mathbb{R}^{1\times d}$, $W \in \mathbb{R}^{d\times h}$, and d, h are positive integers. Let $S = \mathrm{diag}(s_1, \dots, s_d)$ with $s_j > 0$, and let R belong to the orthogonal group $O(d) = \{R \in \mathbb{R}^{d\times d} : R^\top R = I_d\}$, where I_d is the identity matrix. Then, in exact arithmetic,

$$\underbrace{xW}_{\text{original computation}} = \underbrace{(xS^{-1})(SW)}_{\text{redistributed scales}} = \underbrace{(xR)(R^\top W)}_{\text{mixed coordinates}}. \tag{2}$$

Thus, channelwise rescaling and orthogonal changes of coordinates can preserve a full-precision computation while changing its numerical representation. SmoothQuant uses rescaling; QuaRot and SpinQuant use orthogonal transformations (Xiao et al., 2023; Ashkboos et al., 2024; Liu et al., 2025). Following this literature, we call elements of $O(d)$ *rotations*; determinant $+1$ is not required here.

Let $\mathcal{Q}_x$ and $\mathcal{Q}_w$ denote activation and weight quantizers. The corresponding identities generally fail after quantization:

$$\begin{aligned} \mathcal{Q}_x(x)\mathcal{Q}_w(W) &\not\equiv \mathcal{Q}_x(xS^{-1})\mathcal{Q}_w(SW), \\ \mathcal{Q}_x(x)\mathcal{Q}_w(W) &\not\equiv \mathcal{Q}_x(xR)\mathcal{Q}_w(R^\top W). \end{aligned} \tag{3}$$

Here $\not\equiv$ means that equality is not an identity over admissible inputs and parameters, not that it fails at every input. Rounding and clipping depend on the chosen coordinates. Consequently, two representations with the same full-precision output can have different post-quantization outputs. This is the opportunity exploited by equivalence-preserving preprocessing.

Offline preparation and online execution are separate choices. A transformation is *fused* when its effect is absorbed into stored parameters before deployment. An *online* transformation instead acts on activations during inference. Equation equation 2 alone does not establish that an arbitrary rotation can be fused across an entire network: dense rotations generally do not commute with elementwise nonlinearities, and residual branches must use consistent coordinates. Valid rotation-based preprocessing must respect these graph constraints (Ashkboos et al., 2024; Liu et al., 2025); Appendix B.1 gives the relevant identities and their assumptions. Whether online operations are allowed is therefore a deployment constraint, distinct from whether offline parameter changes must preserve the original full-precision function.

Preprocessing and subsequent adaptation are also distinct. Post-training quantization (PTQ) constructs a quantized model from pretrained parameters without conventional end-to-end retraining. It can still optimize scales, rounding decisions, or transformations using calibration data (Nagel et al., 2020; Frantar et al., 2023; Shao et al., 2024). Quantization-aware training (QAT) updates

model parameters while accounting for quantization in the forward pass. Quantization-aware distillation (QAD) uses a frozen high-precision *teacher*'s outputs as targets for the quantized *student* (Hinton et al., 2015; Liu et al., 2024). These choices can be combined. In particular, optimizing a quantized-network objective does not itself require releasing exact equivalence: SpinQuant optimizes rotations while preserving the underlying full-precision computation (Liu et al., 2025). Our question concerns the additional restriction on the *representations being optimized*, not whether quantization appears in the objective.

3 Why Relax Full-Precision Equivalence?

The deployed model is the quantized student, not its unquantized counterpart. This section separates the constraint imposed on the latter from the fidelity sought in the former. Relaxation has two potential benefits: it admits additional candidate representations, and it allows full-precision changes to compensate for quantization error. We first make the comparison precise, then explain which preprocessing choices it permits. None of these arguments requires changing the deployment budget.

3.1 One teacher, two student computations, one objective

Let $\mathcal{X}$ be a nonempty measurable space of admissible token contexts and let $V \geq 2$ be the vocabulary size. The frozen teacher, with parameters θ_0, produces logits $F_0(x) \in \mathbb{R}^V$. A candidate specifies both parameters and how they will be deployed. For a positive integer p, let $\Theta \subseteq \mathbb{R}^p$ be a latent parameter domain containing θ_0. Let Φ be the set of auxiliary preprocessing configurations. For each $\phi \in \Phi$, an offline map $T_\phi : \Theta \to \mathcal{W}_\phi$ produces valid parameters for a specified graph $\mathcal{G}_\phi$; $\mathcal{W}_\phi$ is that graph's parameter domain. A quantization policy $\eta \in \mathcal{E}_\phi$ specifies bit widths, quantization sites, grouping, and scale, clipping, and rounding rules. The set $\mathcal{A}$ consists of the compatible triples $a = (u, \phi, \eta)$, with $u \in \Theta$.

For each candidate, distinguish the following two executions:

$$\begin{aligned} F_a(x) &= F_{\mathcal{G}_\phi}(x; T_\phi(u)) \quad \text{(full-precision student)}, \\ \widehat{F}_a(x) &= F^{Q_\eta}_{\mathcal{G}_\phi}(x; T_\phi(u)) \quad \text{(quantized student)}. \end{aligned} \tag{4}$$

The first map executes the graph without quantization; the second applies the policy Q_η at its designated weight and activation sites. Activation quantization depends on the input, so Q_η is not merely a map on stored weights. The graph includes any retained online transformations. All teacher and student logit maps are assumed measurable and finite-valued on $\mathcal{X}$, with the same output dimension V.

We measure fidelity at the final next-token distribution. For temperature $\tau > 0$, logits $z \in \mathbb{R}^V$, and vocabulary index $v \in \{1, \dots, V\}$, define

$$p_\tau(z)_v = \frac{\exp(z_v/\tau)}{\sum_{w=1}^V \exp(z_w/\tau)}, \qquad \mathrm{KL}(p\|q) = \sum_{v=1}^V p_v \log \frac{p_v}{q_v}. \tag{5}$$

In the second definition, $p, q \in (0,1)^V$ are probability vectors, each summing to one; KL is the Kullback–Leibler divergence, with natural logarithms. Given a nonempty calibration multiset $\mathcal{D} = (x_i)_{i=1}^N$, $N \geq 1$, our empirical distillation objective is

$$\widehat{\mathcal{R}}_{\mathcal{D}}(a) = \frac{\tau^2}{N} \sum_{i=1}^N \mathrm{KL}\Big(p_\tau(F_0(x_i)) \,\Big\|\, p_\tau(\widehat{F}_a(x_i)) \Big). \tag{6}$$

This is the standard temperature-scaled, teacher-to-student distillation direction (Hinton et al., 2015; Liu et al., 2024). Contexts may be prefixes of the same sequence; only valid, non-padding positions enter $\mathcal{D}$. The objective compares the teacher with the *quantized* student. It is neither a tensorwise rounding loss nor a guarantee of downstream accuracy.

3.2 A larger feasible set under the same deployment constraints

Fix a set $\mathcal{A}_B \subseteq \mathcal{A}$ of candidates satisfying deployment constraints B, including permitted graphs, bit widths, and resource budgets. Stored auxiliary tensors count toward storage, and retained online

transformations count toward inference cost. These constraints do not change when we relax full-precision equivalence.

To quantify departure before quantization, fix a probability measure μ on $\mathcal{X}$ and define the mean squared logit drift

$$d_\mu(a) = \frac{1}{V}\mathbb{E}_{x\sim\mu}\big\|F_a(x) - F_0(x)\big\|_2^2, \tag{7}$$

where $\|\cdot\|_2$ is the Euclidean norm. This nonnegative expectation may be $+\infty$. The reference distribution μ specifies where drift is constrained; it need not be the empirical distribution of $\mathcal{D}$. Define

$$\begin{aligned} \mathcal{A}_{\text{eq}} &= \{a \in \mathcal{A}_B : F_a(x) = F_0(x) \text{ for every } x \in \mathcal{X}\}, \\ \mathcal{A}_\epsilon &= \{a \in \mathcal{A}_B : d_\mu(a) \le \epsilon\}, \qquad 0 \le \epsilon < \infty, \\ \mathcal{A}_\infty &= \mathcal{A}_B. \end{aligned} \tag{8}$$

Thus ϵ controls the allowed pre-quantization drift, whereas B controls deployment. Zero drift gives equality only μ-almost surely; it does not automatically give pointwise equivalence on all contexts. Exact logit equality is also stronger than equality of softmax probabilities, which is unchanged by adding a common scalar to all logits. We use the logit-based definition consistently.

Write the best attainable objective values as

$$\widehat{\mathcal{R}}^\star_\epsilon = \inf_{a\in\mathcal{A}_\epsilon} \widehat{\mathcal{R}}_{\mathcal{D}}(a), \qquad \widehat{\mathcal{R}}^\star_{\text{eq}} = \inf_{a\in\mathcal{A}_{\text{eq}}} \widehat{\mathcal{R}}_{\mathcal{D}}(a). \tag{9}$$

We use $\inf \varnothing = +\infty$ and do not assume that an infimum is attained.

Proposition 1 (Relaxation cannot worsen the best attainable objective)**.** *Fix $\mathcal{A}_B$, $\mathcal{D}$, μ, and $\tau > 0$ under the assumptions above. For every $0 \le \epsilon_1 \le \epsilon_2 \le \infty$,*

$$\begin{aligned} &\mathcal{A}_{\text{eq}} \subseteq \mathcal{A}_{\epsilon_1} \subseteq \mathcal{A}_{\epsilon_2}, \\ &\widehat{\mathcal{R}}^\star_{\epsilon_2} \le \widehat{\mathcal{R}}^\star_{\epsilon_1} \le \widehat{\mathcal{R}}^\star_{\text{eq}}. \end{aligned} \tag{10}$$

In words, releasing the equivalence constraint keeps every previously feasible candidate and may admit better ones, without spending a larger deployment budget. This is a statement about feasible sets, not an optimization guarantee: the inequalities need not be strict, and a finite-budget optimizer need not find a better model. Nor does the proposition compare different permitted graph families or establish generalization. Appendix A.2 gives the proof and clarifies the zero-drift case.

3.3 Full-precision drift can compensate quantization error

For a fixed candidate a, define its pre-quantization error relative to the teacher as $e_{\text{FP}}(x) = F_a(x) - F_0(x)$, and the change introduced by quantization as $e_Q(x) = \widehat{F}_a(x) - F_a(x)$. At every context x,

$$\begin{aligned} \|\widehat{F}_a(x) - F_0(x)\|_2^2 &= \|e_{\text{FP}}(x) + e_Q(x)\|_2^2 \\ &= \|e_{\text{FP}}(x)\|_2^2 + \|e_Q(x)\|_2^2 + 2\langle e_{\text{FP}}(x), e_Q(x)\rangle, \end{aligned} \tag{11}$$

where $\langle\cdot,\cdot\rangle$ is the Euclidean inner product. The last term can be negative: a change before quantization can oppose the error introduced by quantization. Enforcing $e_{\text{FP}} = 0$ removes this possibility. Importantly, both errors depend on a; this identity does not assume that quantization error stays fixed as weights change. Appendix A.4 derives the expectation-level identity and bounds under finite-second-moment assumptions.

Equation equation 11 concerns squared logit error, not a decomposition of KL. The following existence result shows that the potential benefit is not restricted to that diagnostic.

Proposition 2 (Strict improvement is possible for the distillation objective)**.** *There exists an instance of the formulation above with a fixed graph, a fixed deterministic quantization policy, a scalar latent parameter, a single calibration context, and $\tau = 1$, for which $\mathcal{A}_{\text{eq}} \neq \varnothing$ and, for some finite $\epsilon > 0$,*

$$\widehat{\mathcal{R}}^\star_\epsilon < \widehat{\mathcal{R}}^\star_{\text{eq}}. \tag{12}$$

In this instance, every exactly equivalent full-precision representation rounds to a worse student than a representation with nonzero drift. Thus, exact equivalence can exclude a better solution even for the same forward-KL objective and the same deployment graph. The constructive proof in Appendix A.5 establishes possibility, not a universal benefit or a causal explanation of transformer results.

3.4 The relaxation permits several preprocessing parameterizations

The formulation is not tied to a particular transformation family. With *auxiliary-only preprocessing*, the base parameters remain $u = \theta_0$ and only (ϕ, η) are optimized. With *joint adaptation*, (u, ϕ, η) may all change. With *direct weight adaptation*, there are no auxiliary preprocessing parameters: set $\Phi = \{\star\}$, $\mathcal{W}_\star = \Theta$, $T_\star(u) = u$, and fix $\mathcal{G}_\star$. Here $\star$ is merely the index of the singleton set. Each case uses Equation equation 6 on the appropriate feasible set; Appendix A.3 writes out the three optimization problems.

An offline map need not be linear or invertible. For example, $T_\phi(u) = u + h_\phi(u)$ may use a nonlinear map $h_\phi : \mathbb{R}^p \to \mathbb{R}^p$, provided the result lies in the valid parameter domain. If only $T_\phi(u)$ is exported, evaluating h_ϕ introduces no inference-time module. At a fixed graph, this cannot enlarge the representable parameter set beyond unrestricted direct weight adaptation; it changes the parameterization of the search. The broader design space therefore concerns *how final parameters are constructed*, not an implicit relaxation of deployment constraints. We next choose a direct-weight realization that retains the closest structural connection to rotation-based preprocessing.

4 Method: Equivalence as Initialization, Not a Constraint

Section 3.4 admits auxiliary, nonlinear, and joint relaxations. To stay as close as possible to prior rotation-based preprocessing, we retain its graph-consistent orthogonal construction and release the restriction that subsequent weights must remain exactly equivalent. Concretely, we use an *exactly equivalent, fixed rotated initialization followed by direct weight adaptation with global QAD*. Rotations select the starting representation; they no longer delimit the representations that optimization may explore. This keeps the rotation-based starting construction while expanding the subsequent search from equivalent representations to the permitted trainable weights.

The distinction from learned-rotation preprocessing is the trainable representation family, not merely the use of a quantized-network objective (Liu et al., 2025). Within the chosen deployment graph, we keep the quantization policy fixed and optimize the latent weights themselves. The method introduces no learned nonlinear preprocessing module and no preprocessing-specific online rotations.

4.1 A fixed, graph-consistent starting point

An orthogonal initializer must preserve the *whole computation*, not just an isolated matrix product. For positive integers d, h and row-vector coordinates, the local identity is

$$xW = (xR)(R^\top W), \qquad x \in \mathbb{R}^{1\times d}, \quad W \in \mathbb{R}^{d\times h}, \quad R \in O(d). \tag{13}$$

Normalization gains, biases, residual branches, attention operations, and any parameter-sharing constraints must be handled consistently. We retain only transformations that can be fused into the chosen graph without changing its full-precision input–output map. Appendix B.1 states sufficient conditions, proves the local identities, and gives the graph-level argument. Exact equivalence here is algebraic; bitwise equality after floating-point fusion is not assumed.

For coordinate mixing, we use a fixed randomized normalized Hadamard construction (Ashkboos et al., 2024; Tseng et al., 2024). Let $K \geq 1$ and $m \geq 0$ be integers with $d = K2^m$. Assume that Hadamard matrices H_K and H_{2^m} with entries in $\{-1, 1\}$ exist and satisfy $H_K H_K^\top = KI_K$ and $H_{2^m} H_{2^m}^\top = 2^m I_{2^m}$. For $j = 1, \ldots, d$, independently sample signs $s_j \in \{-1, 1\}$ with probability $1/2$ each, set $D = \mathrm{diag}(s_1, \ldots, s_d)$, and define

$$R = D\frac{H_K \otimes H_{2^m}}{\sqrt{d}}, \tag{14}$$

where $\otimes$ denotes the Kronecker product. Appendix B.1 proves $R \in O(d)$. The construction preserves Euclidean energy while mixing coordinates; it does not guarantee reduced outliers for every

input. It is data independent and uses no rotation-learning steps. We call this initializer FIXED0: "zero" refers to the number of rotation-learning steps, not to a zero or identity matrix.

Let $\mathcal{G}_0$ be the permitted graph with no preprocessing-specific online rotations, called the *nohad* graph in our comparisons. Let ϕ_0 collect the fixed rotations and their fusion configuration, and let $T_{\text{fuse}}(\theta_0; \phi_0)$ be the resulting offline parameter map into $\mathcal{G}_0$. Under the graph-consistency conditions above, initialization gives

$$u_0 = T_{\text{fuse}}(\theta_0; \phi_0), \qquad F_{\mathcal{G}_0}(x; u_0) = F_0(x) \quad \text{for every } x \in \mathcal{X}. \tag{15}$$

At a site where rotation would require an extra online operation, such as the nonlinear feed-forward activation before a down projection, we omit that rotation *together with its paired weight transformation*. Simply deleting an online operator while keeping its rotated weight would generally break equivalence. All retained preprocessing is absorbed into u_0.

4.2 DIRECT WEIGHT RELAXATION THROUGH GLOBAL DISTILLATION

After initialization, ϕ_0 has served its purpose and is not optimized. Use the direct-weight parameterization $T_\star(u) = u$ with $\mathcal{G}_\star = \mathcal{G}_0$. Fix a quantization policy η_0 and a trainable domain $\Theta_{\text{tr}} \subseteq \Theta$ containing u_0, with any frozen tensors fixed at their initialized values and $(u, \star, \eta_0) \in \mathcal{A}_B$ for every $u \in \Theta_{\text{tr}}$. We target

$$\inf_{u \in \Theta_{\text{tr}}} \widehat{\mathcal{R}}_{\mathcal{D}}(u, \star, \eta_0), \qquad u \text{ initialized at } u_0. \tag{16}$$

No full-precision drift constraint is imposed: this is the $\epsilon = \infty$ specialization, restricted only by the trainable domain and deployment policy. Fixing η_0 fixes the quantization *rules*; scales or clipping bounds computed by those rules may still depend on the current weights and activations.

The forward pass uses *fake quantization*: values are mapped to the prescribed low-bit grid and dequantized for training computation, while the latent weights remain continuous (Jacob et al., 2018). Both weight and activation quantization are included in the student forward pass. "Global" means that Equation equation 6 matches the final next-token distribution, rather than imposing separate reconstruction objectives at individual layers. Updates use a straight-through estimator (STE), a prescribed surrogate backward rule in place of the derivative of rounding (Bengio et al., 2013). This surrogate supports end-to-end weight updates but is not the true gradient of the discontinuous quantized objective. The numerical policy, temperature, optimizer, and training budget are experimental choices specified in Section 5.1; Appendix B.2 details the reported quantizer and surrogate family.

The complete procedure is given in Appendix B.3. It never projects the weights back onto the exactly equivalent family: initialization supplies the starting point, while adaptation is judged by the quantized student's behavior.

Deployment retains the model, not the preprocessing procedure. The teacher, optimizer state, and initialization transforms are discarded at export. The absence of preprocessing-specific online operators does not remove ordinary model operations, quantization scales, or any unquantized tensors. It therefore establishes zero *transformation-specific* overhead relative to $\mathcal{G}_0$, not a measured hardware speedup or compression ratio. Offline training and deployment consume separate resources; Appendix A.6 formalizes this distinction.

5 EXPERIMENTAL RESULTS

5.1 EXPERIMENTAL SETUP

Models & Quantization. We evaluate LLaMA-2-7B, LLaMA-3-8B, Mistral-7B-v0.3, and Qwen2.5-7B under RTN W4A4 quantization (symmetric max-absolute $W4$ with group size 128, per-token asymmetric $A4$, $KV16$, and unquantized LM head) without online Hadamard transforms (see Appendix B.2).

Training. QAD utilizes a frozen FP16 teacher over 800 C4 sequences of length 2,048 (500 AdamW updates, global batch size 64, cosine decay, peak LR 5×10^{-6} to 10^{-5}, forward-KL loss with $\tau = 2$, and identity STE). See Appendix B.2 and B.3 for implementation details.

Table 1: Primary comparison of the local nohad-specific rotation CE control and QAD. The rotation control uses the SpinQuant parameterization but is not an official SpinQuant score: it is trained for 100 updates without online R3/R4 operations under the canonical W4 G128/A4 RTN proxy. QAD uses 800 C4 sequences. All completed cells use the same canonical local RTN W4 G128/A4 evaluator and nohad deployment graph. $\Delta_{\mathrm{PPL}} = \mathrm{PPL}_{\mathrm{rotation}} - \mathrm{PPL}_{\mathrm{QAD}}$, so positive values favor QAD and are shown on the QAD rows.

Model	Method	Updates	PPL ↓	6-task ↑	Δ_{PPL}
LLaMA-2-7B	nohad rotation CE	100	8.4981	0.5827	–
	QAD	500	**7.8436**	**0.6135**	0.6545
LLaMA-3-8B	nohad rotation CE	100	14.6862	0.5434	–
	QAD	500	**12.1814**	**0.6008**	2.5048
Mistral-7B-v0.3	nohad rotation CE	100	8.7973	0.5988	–
	QAD	500	**7.9209**	**0.6364**	0.8764
Qwen2.5-7B	nohad rotation CE	100	15.6718	0.5254	–
	QAD	500	**11.9878**	**0.5958**	3.6840

Scope & Evaluation. We evaluate direct weight adaptation ($\epsilon = \infty$). Perplexity is measured on WikiText-2; zero-shot accuracy is averaged over six standard tasks (ARC-C, ARC-E, HellaSwag, LAMBADA, PIQA, WinoGrande) using standard length-normalization.

Baseline and terminology. The main rotation control uses the SpinQuant parameterization without online Hadamard operations. It is distinct from published SpinQuant scores; Appendix D preserves their source-separated provenance.

5.2 Main Results: Best Converged Solutions

Table 1 compares QAD with a locally optimized, equivalence-preserving rotation control in the nohad graph. QAD improves both perplexity and six-task accuracy across all four model families, with the largest gains on LLaMA-3 and Qwen, where the rotation-only endpoints are also weakest. The consistent direction of improvement across distinct architecture demonstrates that the proposed adaptation is robust to model variation within the evaluated canonical RTN W4A4 setting, rather than benefiting only one architecture or one unusually degraded endpoint. This supports direct weight adaptation as broadly useful way to recover residual quantization error left by fused, equivalence-preserving preprocessing. Section 5.3 complements this cross-model comparison with a controlled equal-budget study of the effect of releasing exact equivalence.

5.3 Ablation: The Effect of Relaxing Exact Equivalence

To isolate the impact of releasing exact equivalence from other training factors, we conduct a controlled, equal-budget ablation study. While Proposition 1 theoretically compares global infima, practical optimization is constrained by finite budgets. Table 2 evaluates whether our relaxed procedure reaches a superior solution compared to a strictly constrained, rotation-only procedure under identical optimization allowances.

Experimental Setup. Both the constrained and relaxed arms are highly controlled to ensure a fair comparison. They start from the same FIXED0 initialization and share the same forward-KL teacher model, C4 training/validation split, training budget (500 updates, 32,000 rank-samples), validation-based checkpoint-selection rule, nohad deployment graph, and canonical RTN W4 G128/A4 evaluator. The only difference (intervention) is the optimization search space:

- **Constrained FKL** optimizes graph-consistent Cayley rotations that preserve the full-precision function exactly.
- **Relaxed FKL** directly optimizes the latent weights without any drift constraints.

The metric $\Delta_{\mathrm{space}} = \mathrm{PPL}_{\mathrm{constrained\ FKL}} - \mathrm{PPL}_{\mathrm{relaxed\ FKL}}$ quantifies the finite-budget advantage of releasing the equivalence constraint. Both checkpoints are selected on disjoint C4 validation data prior to final evaluation.

Table 2: Finite-budget comparison of selected rotation-only and direct-weight parameterizations. "Constrained FKL" and "Relaxed FKL" share the FIXED0 initialization, C4 data, forward-KL loss, optimization budget, validation-only checkpoint selection, deployment graph, and canonical RTN W4 G128/A4 evaluator. "Local nohad rotation CE" uses a different objective and 100-update protocol and is not part of Δ_{space}.

Model	Online Had	Method	Updated	PPL ↓	6-task ↑	Δ_{space}
LLaMA-2-7B	no	FIXED0 PTQ	none	19.4701	0.4295	–
	no	constrained FKL	rotations	8.8896	0.5940	–
	no	relaxed FKL	weights	**7.8436**	**0.6135**	1.0460
	no	nohad rotation CE	rotations	8.4981	0.5827	–
LLaMA-3-8B	no	FIXED0 PTQ	none	59.2573	0.4308	–
	no	constrained FKL	rotations	15.9817	0.5429	–
	no	relaxed FKL	weights	**12.1814**	**0.6008**	3.8003
	no	nohad rotation CE	rotations	14.6862	0.5434	–

Main Findings. Under this matched procedure, direct-weight relaxation consistently outperforms constrained rotation optimization. As shown in Table 2, relaxed FKL improves both perplexity and six-task accuracy for both LLaMA-2 and LLaMA-3, with a particularly pronounced margin on LLaMA-3 ($\Delta_{\text{space}} = 3.80$). Since all other hyperparameters, data, and initialization are strictly controlled, this consistent advantage provides direct evidence that relaxing exact equivalence grants the optimizer access to significantly more effective quantized solutions. Furthermore, replicating this performance improvement across both LLaMA-2 and LLaMA-3 demonstrates that the benefits of relaxed parameterization are robust and model-agnostic under the evaluated W4A4 settings.

Practical Comparison. To provide a broader context, the "nohad rotation CE" row serves as an additional rotation-only baseline. This baseline utilizes the SpinQuant parameterization with a cross-entropy objective and a 100-update protocol. While it is trained under a different objective, it highlights that relaxed FKL achieves superior downstream performance even compared to alternative rotation-only training strategies.

Additional results on alternative initializations, longer training trajectories, lower-bit settings, and online transforms are provided in Appendix C.

6 LIMITATIONS

Our evidence is limited to four decoder-only models in the 7–8B range, English-language calibration and evaluation data, and one primary W4A4 RTN recipe. WikiText-2 perplexity and six zero-shot tasks do not cover larger or mixture-of-experts models, multilingual and long-context settings, instruction following, safety, or interactive generation. we do not report hardware-specific benchmarks such as latency, memory footprint, or throughput.

Additionally, our optimization results are subject to finite computational budgets and specific parameterizations. QAD requires a full-precision teacher model and substantial offline multi-GPU resources. Consequently, we demonstrate the potential benefit of relaxing exact equivalence within our tested configurations rather than claiming universal optimality across all optimization regimes.

7 CONCLUSION

We reframed quantization preprocessing as a search over deployable quantized models rather than only full-precision-equivalent representations. A deterministic structured initialization followed by global quantization-aware distillation improves nohad W4A4 endpoints across four model families and outperforms a matched finite-budget rotation-only procedure on LLaMA-2 and LLaMA-3. Exact equivalence remains a useful initialization principle, but our results show that enforcing it throughout optimization can be unnecessarily restrictive.

AI USE STATEMENT

We used generative AI assistants for writing polish, literature retrieval, research ideation and execution (including coding and debugging), and drafting sections of the manuscript. All core research questions, experimental designs, and substantive claims were determined solely by the authors, who carefully reviewed all AI-assisted outputs and take full responsibility for the paper's content and artifacts.

REPRODUCIBILITY STATEMENT

We provide an anonymized artifact containing complete code for training, quantization, and evaluation, along with environment specifications, configuration templates, and execution scripts. Key experimental settings—including the quantizer, calibration data, and optimization budgets—are detailed in the paper and configuration files. Model checkpoints and datasets are not redistributed and must be obtained under their original access conditions. Full training requires CUDA-capable multi-GPU hardware, while command generation and static checks can run without downloading models.

ETHICS STATEMENT

This work focuses on numerical compression of existing language models and utilizes public datasets, benchmarks, and pretrained checkpoints, which may contain social bias, offensive content. While quantization reduces deployment costs and broadens model accessibility, it can also lower the barriers to misuse and does not inherently mitigate pre-existing safety risks. Users remain responsible for safety evaluations and licensing compliance.

REFERENCES

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

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

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. EfficientQAT: Efficient quantization-aware training for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10081–10100. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.498. URL https://aclanthology.org/2025.acl-long.498/.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*, 2022. URL https://arxiv.org/abs/2208.07339. NeurIPS 2022; arXiv title used (proceedings title: GPT3.int8()).

DaYou Du, Yijia Zhang, Shijie Cao, Jiaqi Guo, Ting Cao, Xiaowen Chu, and Ningyi Xu. BitDistiller: Unleashing the potential of sub-4-bit LLMs via self-distillation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 102–116. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.7. URL https://aclanthology.org/2024.acl-long.7/.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.17323.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2704–2713, 2018. doi: 10.1109/CVPR.2018.00286. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Jacob_Quantization_and_Training_CVPR_2018_paper.html.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. LLM-QAT: Data-free quantization aware training for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 467–484. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.26. URL https://aclanthology.org/2024.findings-acl.26/.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. SpinQuant: LLM quantization with learned rotations. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2405.16406v4. Camera-ready arXiv version 4; ICLR 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7197–7206. PMLR, 2020. URL https://proceedings.mlr.press/v119/nagel20a.html.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. OmniQuant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2308.13137.

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*, pp. 48630–48656. PMLR, 2024. URL https://proceedings.mlr.press/v235/tseng24a.html.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38087–38099. PMLR, 2023. URL https://proceedings.mlr.press/v202/xiao23c.html.

Appendix Overview

This appendix contains the following sections:

- **Section A**: Definitions and Mathematical Details
- **Section B**: Graph Consistency and the Evaluated Training Recipe
- **Section C**: Additional Experimental Results
- **Section D**: Baseline Provenance
- **Section E**: Exploratory Diagnostics and Their Interpretation

A Definitions and Mathematical Details

This appendix proves the mathematical statements used in Sections 3 and 4. The constructive example is an analytical instance of the formulation, not an additional language-model experiment.

A.1 Notation and well-posedness

Table 3: Core notation. The quantized execution includes any designated input-dependent activation quantizers.

Symbol	Meaning
$\mathcal{X}$, V	Measurable context space and vocabulary size, $V \geq 2$.
θ_0, F_0	Frozen teacher parameters and next-token logit map.
$\Theta \subseteq \mathbb{R}^p$, u	Latent parameter domain, $p \geq 1$, and latent parameters.
Φ, ϕ	Auxiliary preprocessing domain and configuration.
T_ϕ, $\mathcal{W}_\phi$	Offline parameter map and graph-compatible output domain.
$\mathcal{G}_\phi$	Specified inference graph, including any online transformations.
$\mathcal{E}_\phi$, η, Q_η	Allowed quantization policies, a policy, and its execution rules.
a, $\mathcal{A}$	Compatible triple (u, ϕ, η) and ambient candidate set.
B, $\mathcal{A}_B$	Fixed deployment constraints and candidates satisfying them.
F_a, $\widehat{F}_a$	Full-precision and quantized student logit maps.
$\mathcal{D}$, N, τ	Calibration multiset, its size $N \geq 1$, and temperature $\tau > 0$.
p_τ, KL	Temperature-scaled softmax and Kullback–Leibler divergence.
μ, d_μ	Reference probability measure and full-precision logit drift.
$\mathcal{A}_{\mathrm{eq}}$, $\mathcal{A}_\epsilon$	Pointwise equivalent and drift-relaxed feasible sets.
$\widehat{\mathcal{R}}_{\mathcal{D}}$	Empirical teacher-to-quantized-student forward-KL objective.
$\widehat{\mathcal{R}}^\star_{\mathrm{eq}}$, $\widehat{\mathcal{R}}^\star_\epsilon$	Infima of that objective over the respective feasible sets.
e_{FP}, e_Q	Teacher-to-FP-student error and the change due to quantization.
q_μ, r_μ	Mean squared self-quantization and teacher-to-quantized errors.
$\star$	Singleton index for a parameterization with no auxiliary parameters.
$\mathcal{G}_0$, u_0, η_0	Chosen no-online-rotation graph, initialization, and fixed policy.
Θ_{tr}	Admissible trainable domain with any frozen tensors held fixed.

A discrete context space may be equipped with its power-set sigma-algebra. More generally, all logit maps are assumed measurable with respect to the chosen sigma-algebra and finite-valued on $\mathcal{X}$. For finite logits and $\tau > 0$, every softmax probability is strictly positive. Hence the finite sum in Equation equation 6 is a finite real number for each candidate. Its nonnegativity follows from $\log t \leq t - 1$ for $t > 0$: for positive probability vectors p, q,

$$-\mathrm{KL}(p\|q) = \sum_v p_v \log(q_v/p_v) \leq \sum_v p_v(q_v/p_v - 1) = 0.$$

Nonnegative expectations such as d_μ are well defined in $[0, +\infty]$. We use $\inf \varnothing = +\infty$ and assume neither compactness nor continuity through rounding nor attainment of an infimum. A numerical computation returning NaN is a failed computation, not a candidate with a valid extended-real objective value.

The definition $\mathcal{A}_\infty = \mathcal{A}_B$ includes candidates with infinite drift; it is not defined as the union of the finite-drift feasible sets. The objective data $\mathcal{D}$, the drift distribution μ, and the temperature τ are fixed throughout each feasible-set comparison.

Exact preservation means equality of logit vectors, not merely next-token probabilities. Adding a context-dependent scalar to every logit leaves softmax probabilities unchanged because the common exponential factor cancels between numerator and denominator. A probability-based relaxation could be defined instead, but its drift and equality set would have to be changed together. The set-inclusion argument below requires only a nonnegative drift functional that vanishes on the exact-preserving set.

A.2 Feasible-set nesting and what it does not prove

Proof of Proposition 1. If $a \in \mathcal{A}_{\text{eq}}$, then $F_a(x) - F_0(x) = 0$ for every $x \in \mathcal{X}$. The drift integrand is therefore identically zero, so $d_\mu(a) = 0$ and $a \in \mathcal{A}_0$. For finite $0 \le \epsilon_1 \le \epsilon_2$, the inequality $d_\mu(a) \le \epsilon_1$ implies $d_\mu(a) \le \epsilon_2$. This proves the two inclusions when both bounds are finite. If $\epsilon_2 = \infty$, the second inclusion follows from $\mathcal{A}_{\epsilon_1} \subseteq \mathcal{A}_B = \mathcal{A}_\infty$; if also $\epsilon_1 = \infty$, it is equality. The first inclusion follows in every case from $\mathcal{A}_{\text{eq}} \subseteq \mathcal{A}_0 \subseteq \mathcal{A}_{\epsilon_1}$.

For sets $S \subseteq T$ and a fixed objective f, $\inf_T f \le f(a)$ for every $a \in S$, hence $\inf_T f \le \inf_S f$ when S is nonempty. When S is empty the inequality holds under $\inf \varnothing = +\infty$. Apply this fact to the two inclusions, using $f = \widehat{\mathcal{R}}_{\mathcal{D}}$, to obtain Equation equation 10. □

For completeness, $d_\mu(a) = 0$ implies $F_a = F_0$ only μ-almost surely. Indeed, set $Z(x) = \|F_a(x) - F_0(x)\|_2^2$. For each positive integer n, $\mathbb{E}_\mu Z \ge \mu(Z \ge 1/n)/n$. Zero expectation forces all these probabilities to vanish; since $\{Z > 0\} = \bigcup_{n \ge 1} \{Z \ge 1/n\}$, $Z = 0$ almost surely. Conversely, almost-sure equality implies zero drift. On a countable context space with positive probability at every context, this is pointwise equality. An empirical reference distribution constrains only its observed contexts.

The inequalities can be equalities, for example when an exact-preserving candidate already minimizes the quantized objective. A larger feasible set also need not yield a better candidate under a fixed-budget optimization procedure. The proposition holds the permitted quantizer and graph families fixed; it does not compare an online-transform graph against a graph that forbids those transformations, or establish held-out perplexity or downstream accuracy.

A.3 Auxiliary-only, joint, and direct optimization

For a fixed teacher $\theta_0 \in \Theta$, auxiliary-only preprocessing solves

$$\inf_{\substack{\phi \in \Phi,\ \eta \in \mathcal{E}_\phi \\ (\theta_0, \phi, \eta) \in \mathcal{A}_\epsilon}} \widehat{\mathcal{R}}_{\mathcal{D}}(\theta_0, \phi, \eta). \tag{17}$$

The latent base parameters are frozen, but the exported parameters $T_\phi(\theta_0)$ can still change. An equivalence-preserving rotation family restricts ϕ to compatible orthogonal transformations and additionally imposes membership in $\mathcal{A}_{\text{eq}}$. A fixed-rotation method chooses a single ϕ and need not optimize Equation equation 17.

Allowing both latent parameters and preprocessing parameters to change gives

$$\inf_{\substack{u \in \Theta,\ \phi \in \Phi,\ \eta \in \mathcal{E}_\phi \\ (u, \phi, \eta) \in \mathcal{A}_\epsilon}} \widehat{\mathcal{R}}_{\mathcal{D}}(u, \phi, \eta). \tag{18}$$

For direct adaptation with no auxiliary parameters, set $\Phi = \{\star\}$, $\mathcal{W}_\star = \Theta$, $T_\star(u) = u$, and fix the graph $\mathcal{G}_\star$. The problem is then

$$\inf_{\substack{u \in \Theta,\ \eta \in \mathcal{E}_\star \\ (u, \star, \eta) \in \mathcal{A}_\epsilon}} \widehat{\mathcal{R}}_{\mathcal{D}}(u, \star, \eta). \tag{19}$$

A fixed quantization policy corresponds to a singleton $\mathcal{E}_\phi = \{\eta_0\}$. Learning clipping thresholds or other quantizer parameters instead leaves the corresponding coordinates of η free. Fixing a policy does not require the scales that it computes to be independent of current weights or activations.

An offline residual parameter map can have the form $T_\phi(u) = u + h_\phi(u)$, where $h_\phi : \mathbb{R}^p \to \mathbb{R}^p$ is nonlinear and its output is restricted so that $T_\phi(u)$ is a valid parameter vector. If only this vector is exported, h_ϕ is not evaluated on future tokens. For a fixed graph with parameter domain Θ, any such image satisfies $T_\phi(\Theta) \subseteq \Theta$. Thus it cannot represent weights unavailable to unrestricted direct optimization over Θ; it changes the parameterization, restrictions, or optimization bias. This concerns nonlinear *parameter preparation*, not an added inference-time nonlinearity.

A.4 How departure from the teacher interacts with quantization

Fix a candidate a and use the error vectors defined in Section 3.3. Assume

$$\mathbb{E}_\mu \|e_{\text{FP}}(x)\|_2^2 < \infty, \qquad \mathbb{E}_\mu \|e_Q(x)\|_2^2 < \infty.$$

Define the student's self-quantization discrepancy and its quantized discrepancy from the teacher, respectively, by

$$q_\mu(a) = \frac{1}{V}\mathbb{E}_\mu\|e_Q(x)\|_2^2, \qquad r_\mu(a) = \frac{1}{V}\mathbb{E}_\mu\|\widehat{F}_a(x) - F_0(x)\|_2^2. \tag{20}$$

Then

$$r_\mu(a) = d_\mu(a) + q_\mu(a) + \frac{2}{V}\mathbb{E}_{x\sim\mu}\langle e_{\mathrm{FP}}(x), e_Q(x)\rangle, \tag{21}$$

and

$$\left(\sqrt{d_\mu(a)} - \sqrt{q_\mu(a)}\right)^2 \le r_\mu(a) \le \left(\sqrt{d_\mu(a)} + \sqrt{q_\mu(a)}\right)^2. \tag{22}$$

Derivation of Equations equation 11, equation 21, and equation 22. For every x, the definitions give $\widehat{F}_a(x) - F_0(x) = e_{\mathrm{FP}}(x) + e_Q(x)$. Expanding the Euclidean inner product of this sum with itself proves Equation equation 11. The finite-second-moment assumptions imply that the squared norm of the sum is integrable, since it is at most $2\|e_{\mathrm{FP}}\|_2^2 + 2\|e_Q\|_2^2$. They also ensure that the cross term is absolutely integrable: applying Cauchy–Schwarz first to the vectors and then to scalar random variables gives

$$\mathbb{E}_\mu\left|\langle e_{\mathrm{FP}}, e_Q\rangle\right| \le \mathbb{E}_\mu(\|e_{\mathrm{FP}}\|_2\|e_Q\|_2) \le V\sqrt{d_\mu(a)q_\mu(a)}.$$

Taking expectations in the pointwise expansion and dividing by V is therefore valid and proves Equation equation 21. Bounding the cross term above and below by $2\sqrt{d_\mu(a)q_\mu(a)}$ in absolute value gives Equation equation 22. □

A negative cross term reduces the combined error relative to the sum of its two squared components. Neither this identity nor its bounds guarantee that changing a candidate reduces r_μ, because all terms depend on that candidate. These are squared-logit relations, not KL decompositions. They also do not require the self-quantization discrepancy q_μ to decrease whenever teacher fidelity improves.

A.5 A CONSTRUCTIVE PROOF OF STRICT IMPROVEMENT IN FORWARD KL

Proof of Proposition 2. Take $\mathcal{X} = \{x_\circ\}$ with its power-set sigma-algebra, $\mu(\{x_\circ\}) = 1$, $\mathcal{D} = (x_\circ)$, and $V = 2$. Let $\Theta = [-1, 1] \subset \mathbb{R}$, $\Phi = \{\star\}$, $T_\star(u) = u$, and fix one graph with full-precision logits

$$F_u(x_\circ) = (u^2, 0).$$

Here F_u abbreviates $F_{(u,\star,\eta_0)}$ for the fixed policy η_0 defined next. Quantize the scalar weight using $q(u) = \mathrm{round}(u) \in \{-1, 0, 1\}$, with a fixed rule at half-integer ties, and leave all other operations unchanged. The quantized logits are $\widehat{F}_u(x_\circ) = (q(u)^2, 0)$. All candidates share this graph and policy, and set $\mathcal{A}_B = \{(u, \star, \eta_0) : u \in [-1, 1]\}$. The maps are measurable and finite; no ties occur at the particular weights below.

Choose teacher weight $\theta_0 = 3/5$, so $F_0(x_\circ) = (9/25, 0)$. Exact preservation requires $u^2 = 9/25$, hence precisely $u = \pm 3/5$. Both quantize to a student with logits $(1, 0)$. The relaxed weight $u = 2/5$ instead has logits $(4/25, 0)$ before quantization and $(0, 0)$ afterward. Its drift is

$$d_\mu(2/5, \star, \eta_0) = \frac{1}{2}\left(\frac{4}{25} - \frac{9}{25}\right)^2 = \frac{1}{50}.$$

It therefore belongs to $\mathcal{A}_{1/50}$ but not $\mathcal{A}_{\mathrm{eq}}$.

Set $\tau = 1$ and let $\pi = 1/(1+\exp(-9/25))$ be the teacher's probability for the first vocabulary item. Define $L(z)$ as its forward KL to student probabilities with logits $(z, 0)$. By direct substitution into Equation equation 5,

$$L(z) = \pi\log\pi + (1-\pi)\log(1-\pi) + \log(1+e^z) - \pi z.$$

Consequently,

$$L(1) - L(0) = \log\left(\frac{1+e}{2}\right) - \pi > 0. \tag{23}$$

To verify the strict sign without numerical approximation, observe that for $0 \le t < 3$,

$$e^t = 1 + t + \sum_{n=2}^{\infty} \frac{t^n}{n!} \le 1 + t + \frac{t^2}{2(1 - t/3)}.$$

The inequality follows from $n! \ge 2 \cdot 3^{n-2}$ for $n \ge 2$, which holds at $n = 2$ and is preserved when multiplying by $n + 1 \ge 3$. Applying the bound at $t = 9/25$ yields $e^{9/25} \le 1577/1100 < 3/2$, so $\pi < 3/5$. Applying it at $t = 3/5$ gives $e^{3/5} \le 73/40 < 11/6$. Meanwhile, the positive exponential series gives $e > 1 + 1 + 1/2 + 1/6 = 8/3$, hence $(1 + e)/2 > 11/6$. Since the logarithm is strictly increasing, $\log((1 + e)/2) > 3/5 > \pi$, proving Equation equation 23.

Every exactly equivalent candidate has objective $L(1)$, whereas the candidate at $u = 2/5$ has objective $L(0)$. Thus

$$\widehat{\mathcal{R}}^{\star}_{1/50} \le L(0) < L(1) = \widehat{\mathcal{R}}^{\star}_{\mathrm{eq}},$$

with a nonempty exactly equivalent family, as required. □

This construction proves a strict relaxation benefit for the same forward-KL objective, not a guarantee for transformers. It also does not attribute that benefit to a negative cross term in Equation equation 21: for the selected relaxed weight, the two error vectors point in the same direction. The compensation identity and the strict-improvement example establish distinct, compatible reasons not to equate full-precision preservation with post-quantization fidelity.

A.6 Training resources and deployment resources

Let $M(a)$ be stored model bytes, $C_{\mathrm{on}}(a)$ a specified transformation-operation cost, and $\overline{M}, \overline{C} \ge 0$ their permitted budgets. One possible deployment set is

$$\mathcal{A}_B = \{a \in \mathcal{A} : M(a) \le \overline{M},\ C_{\mathrm{on}}(a) \le \overline{C},\ \mathcal{G}_\phi \in \mathfrak{G}_B\}, \tag{24}$$

where $\mathfrak{G}_B$ is the allowed graph family. A hardware latency constraint would require a stated device and workload, rather than treating operation count as latency. Stored scales and any retained auxiliary tensors belong in M.

Training cost depends on an optimization procedure and its trajectory. Let Π be a procedure taking $(\theta_0, \mathcal{D})$ to a final candidate a_Π, and $C_{\mathrm{train}}(\Pi; \theta_0, \mathcal{D})$ its measured resource cost under a stated accounting convention. For a prescribed nonnegative training budget $\overline{C}_{\mathrm{train}}$, a budgeted procedure must satisfy $C_{\mathrm{train}}(\Pi; \theta_0, \mathcal{D}) \le \overline{C}_{\mathrm{train}}$. This is separate from optimizing over final candidates. Adding an unweighted number of training seconds or inference operations to KL has no intrinsic meaning; a penalized formulation would require explicit conversion weights and would still not make the resources interchangeable.

B Graph Consistency and the Evaluated Training Recipe

B.1 Valid fusion and the boundary imposed by nonlinearities

All identities in this subsection concern exact real arithmetic. We first state the local conditions required by the initializer and then explain why their consistent composition preserves the complete logit map.

Affine maps and residual addition. Let $x \in \mathbb{R}^{1\times d}$, $W \in \mathbb{R}^{d\times h}$, $b \in \mathbb{R}^{1\times h}$, $R \in O(d)$, and $S \in O(h)$. With $y = xW + b$, the transformed parameters satisfy

$$(xR)(R^\top W S) + bS = (xW + b)S = yS. \tag{25}$$

This follows by associativity and $RR^\top = I_d$. Taking $S = I_h$ recovers the input-side identity. Biases therefore transform with output coordinates, not with input coordinates. For $x, y \in \mathbb{R}^{1\times d}$, linearity gives $(x + y)R = xR + yR$; both residual branches must have the same R. An embedding table $E \in \mathbb{R}^{V\times d}$ can similarly be replaced by ER: every selected embedding row is then expressed in the rotated coordinates. If a parameter is shared by multiple operations, all required transformations of that parameter must agree; otherwise that rotation is not an admissible fused initialization.

Root-mean-square normalization. For $\delta > 0$, define gain-free root-mean-square normalization by $r_\delta(x) = x/\sqrt{\|x\|_2^2/d + \delta}$. Since $\|xR\|_2^2 = xRR^\top x^\top = \|x\|_2^2$,

$$r_\delta(xR) = r_\delta(x)R. \tag{26}$$

For a learned gain vector $g \in \mathbb{R}^d$, the following affine projection is $r_\delta(x)\,\mathrm{diag}(g)W + b$. Absorb the gain into $\widetilde{W} = \mathrm{diag}(g)W$ before rotating; then $r_\delta(xR)(R^\top\widetilde{W}) + b = r_\delta(x)\widetilde{W} + b$. Every consumer of the normalization output must be treated consistently. This proof does not assume that arbitrary gains commute with R, or that every other normalization operator is rotationally equivariant.

Attention operations. Let n, r, h be positive integers, let $P \in \mathbb{R}^{n\times n}$ be a fixed attention-weight matrix, let $Z \in \mathbb{R}^{n\times r}$ be the value activations of one head, and let $W \in \mathbb{R}^{r\times h}$ be its output projection. For $U \in O(r)$,

$$(P(ZU))(U^\top W) = (PZ)W. \tag{27}$$

This is again associativity and $UU^\top = I_r$. The condition that P is unchanged matters: rotating values does not by itself justify changing queries or keys. For query and key matrices $A, B \in \mathbb{R}^{n\times r}$, a common rotation at the dot-product inputs satisfies $(AU)(BU)^\top = AB^\top$. If a positional operation intervenes between a projection and this dot product, moving the rotation across it requires an additional commutation identity. We do not assume that identity for an arbitrary positional operation. Biases and shared value heads must obey the same consistency conditions as other affine maps and shared parameters.

Why nonlinearities restrict fusion. Applying an elementwise nonlinearity before a rotation generally differs from applying it afterward. For example, take $\psi(t) = \max\{t, 0\}$ entrywise, $x = (1, -1)$, and

$$R = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}.$$

Then $\psi(xR) = (\sqrt{2}, 0)$ whereas $\psi(x)R = (1/\sqrt{2}, 1/\sqrt{2})$. A valid construction can instead compensate the rotated residual input in the preceding affine map, leave the nonlinear intermediate coordinates unchanged, and rotate the residual output through the final affine map. This does not require commuting a rotation through the nonlinearity. Gated products likewise require a valid local identity; orthogonality alone is insufficient.

Sufficient graph-level conditions. Assume that, after grouping any absorbed gains with their affine consumers, the original and transformed computations can be represented on the same finite directed acyclic graph. On each feature edge e, choose a channel dimension d_e and $R_e \in O(d_e)$; the intended coordinate change is $z_e \mapsto z_e R_e$, applied rowwise to matrix-valued activations. Inputs without a feature coordinate, such as token indices or masks, are unchanged, and external logits have the identity coordinate change. For each node v, let f_v and f'_v be its original and transformed operations. Require, for all admissible inputs,

$$f'_v(z_1R_{e_1}, \ldots, z_kR_{e_k}) = f_v(z_1, \ldots, z_k)R_{e_0}, \tag{28}$$

where $e_1, \ldots, e_k$ are its feature input edges and e_0 is its output edge; any unchanged non-feature inputs are suppressed from this notation. For a node with several outputs, impose this condition on each output. Finally, require that the transformed operations are realizable by valid parameters in $\mathcal{G}_0$, satisfy all parameter-sharing constraints, and introduce no forbidden online operations. These are sufficient conditions on the initializer, not an assertion that every assignment of orthogonal matrices admits fusion.

Graph-level preservation in Equation equation 15. Order the nodes topologically, which is possible because the graph is finite and acyclic. External inputs agree in the two computations. At an embedding lookup, the replacement $E \mapsto ER_e$ produces the required rotated feature row. Inductively, assume that each input to a node equals its original value in the prescribed coordinates. Equation equation 28 then ensures the same property for each output of that node. The affine, residual, normalization, and attention identities above provide valid local steps whenever their stated hypotheses hold; other operations must satisfy the same local condition or remain in unchanged coordinates. The induction reaches every node. At the external logit output, the coordinate change is the identity, so $F_{\mathcal{G}_0}(x; T_{\mathrm{fuse}}(\theta_0; \phi_0)) = F_0(x)$ for every admissible x. This proves the claimed full-precision equivalence under the stated conditions. □

Hadamard orthogonality.

Orthogonality of Equation equation 14. Use the factors and dimensions from Section 4.1, and set $H = H_K \otimes H_{2^m}$. The Kronecker product identity gives

$$HH^\top = (H_K H_K^\top) \otimes (H_{2^m} H_{2^m}^\top) = (KI_K) \otimes (2^m I_{2^m}) = dI_d.$$

Each diagonal entry of D has square one, so $DD^\top = D^\top D = I_d$. Therefore

$$RR^\top = \frac{DHH^\top D^\top}{d} = I_d.$$

Since R is square, this also implies $R^\top R = I_d$, proving $R \in O(d)$. In particular, $\|xR\|_2 = \|x\|_2$ for every row vector x. Orthogonality does not force determinant $+1$, and does not guarantee a reduction in outliers for each individual input distribution. □

Removing an online transform. For an intermediate activation $z \in \mathbb{R}^{1\times d}$, $W \in \mathbb{R}^{d\times h}$, and $H \in O(d)$, the online representation $(zH)(H^\top W)$ equals zW. Deleting only the activation transformation leaves $zH^\top W$, which is generally different. A graph without that online transform must instead restore W or provide another graph-consistent compensation. This is why the initialization omits an inadmissible rotation together with its paired weight transformation, rather than deleting an operator from an otherwise unchanged rotated model.

B.2 Quantization family and straight-through differentiation

For a scalar $t \in \mathbb{R}$, scale $s > 0$, integer zero point z, and integer code bounds $q_{\min} < q_{\max}$, a dequantized uniform quantizer has the form

$$Q(t; s, z) = s\big[\mathrm{clip}_{[q_{\min}, q_{\max}]}(\mathrm{round}(t/s) + z) - z\big], \tag{29}$$

where $\mathrm{clip}_{[a,b]}(v) = \min\{b, \max\{a, v\}\}$ and round rounds to the nearest integer under a fixed tie rule. Tensor quantization applies the rule entrywise with scales and zero points shared according to a stated grouping. An unsigned asymmetric b-bit code can use $q_{\min} = 0$, $q_{\max} = 2^b - 1$, with b a positive integer. A symmetric code sets $z = 0$ and must specify its signed endpoints. The canonical W4 policy uses groups of 128 weights, $z = 0$, $q_{\min} = -7$, $q_{\max} = 7$, and max-absolute scale $s = \max|t|/7$, without clipping search; A4 uses per-token asymmetric codes in $[0, 15]$. Historical results with other mappings are labeled separately. At one bit, the implementation instead uses binary signs with a per-group mean-absolute scale.

The scale/zero-point form in Equation equation 29 is standard (Jacob et al., 2018). For fixed scale, zero point, and bounds, the derivative of Equation equation 29 is zero almost everywhere and undefined at its jumps. It is not the derivative used by QAD. The primary experiments use the identity STE. For a normalized pre-rounding coordinate $v_j = u_j/s$, the implementation also provides the optional surrogate multiplier

$$g_\tau(v_j) = \mathrm{sech}^2\left(\frac{v_j - \mathrm{round}(v_j)}{\tau}\right), \quad \tau > 0, \qquad \mathrm{sech}(t) = \frac{2}{e^t + e^{-t}}, \tag{30}$$

which multiplies the gradient through rounding and attenuates it as the residual from the nearest integer grows. It is a backward-pass surrogate, not the derivative of rounding. Setting the smoothing option to zero selects the identity STE instead. The STE citation (Bengio et al., 2013) supports the surrogate-gradient principle, not this particular optional formula.

Reported stabilization variants. A confidence-adaptive forward/reverse-KL mixture can be represented by

$$\mathcal{L}_{\mathrm{CAKL}} = \frac{\tau^2}{N} \sum_{i=1}^{N} \left[\lambda_i \,\mathrm{KL}(p_i^T \| p_i^S) + (1 - \lambda_i)\,\mathrm{KL}(p_i^S \| p_i^T)\right], \tag{31}$$

where $p_i^T = p_\tau(F_0(x_i))$, $p_i^S = p_\tau(\widehat{F}_a(x_i))$, and $\lambda_i \in [0, 1]$ is confidence derived. In the reported CAKL implementation, one minibatch-level coefficient is used: $\lambda = 1 - \gamma$, where γ is the mean teacher top-1 probability over tokens; gradients do not flow through γ. Confidence-aware KL has a

Algorithm 1 Fixed equivalent initialization followed by direct-weight QAD

Require: Frozen teacher F_0 with parameters θ_0; nonempty calibration multiset $\mathcal{D}$; graph-consistent initialization configuration ϕ_0; graph $\mathcal{G}_0$; quantization policy η_0; trainable domain Θ_{tr}; temperature $\tau > 0$; positive integer step count J; prescribed optimizer, surrogate backward, and checkpoint-selection rules.

Construct $u_0 = T_{\mathrm{fuse}}(\theta_0; \phi_0) \in \Theta_{\mathrm{tr}}$ and initialize optimizer state ω_0.
Fix $\mathcal{G}_\star = \mathcal{G}_0$, $T_\star(u) = u$, and η_0.
for $t = 0, \ldots, J-1$ **do**
    Obtain a nonempty minibatch $\mathcal{D}_t$ of valid contexts from $\mathcal{D}$.
    Evaluate F_0 and the fake-quantized student on $\mathcal{D}_t$.
    Compute $L_t = \widehat{\mathcal{R}}_{\mathcal{D}_t}(u_t, \star, \eta_0)$ using Equation equation 6.
    Compute surrogate gradient g_t using the prescribed backward rule.
    Update (u_t, ω_t) to (u_{t+1}, ω_{t+1}), keeping $u_{t+1} \in \Theta_{\mathrm{tr}}$.
end for
Select a latent checkpoint using the prescribed rule.
Export its quantized weights and the activation-quantization policy for $\mathcal{G}_0$.

precedent in BitDistiller (Du et al., 2024). The standard FKL result ($\lambda = 1$) remains distinct from the CAKL rows.

For latent iterates u_t, a parameter EMA has recursion $\bar{u}_t = \rho \bar{u}_{t-1} + (1-\rho)u_t$ with reported decay $\rho = 0.99$ and optimization-step index $t \geq 1$. The averaged tensor subset, initialization $\bar{u}_0$, update timing, and final export/checkpoint rule must be stated by the implementation. One must quantize the selected exported parameters: quantization of an average generally differs from averaging quantized parameters.

B.3 A PRECISE ABSTRACT PROCEDURE, NOT REPLACEMENT TRAINING CODE

Algorithm 1 gives the method independently of numerical settings. For the experimental configuration in Section 5.1, the standard procedure is as follows. Compute $u_0 = T_{\mathrm{fuse}}(\theta_0; \phi_0)$ and remove all initialization-only transformation objects from the inference graph. At each training step, evaluate the frozen teacher and the fake-quantized student on the same valid contexts, compute the forward-KL loss, backpropagate using the specified surrogate, and apply the configured AdamW update to trainable latent tensors. After the prescribed checkpoint-selection rule, export the selected latent tensors through the quantizer into the same nohad graph. EMA and CAKL are separate optional substitutions, not implicit parts of this standard recipe.

An implementation must check full-precision teacher/initialized-student agreement on representative inputs at a stated numerical tolerance, validate that no unintended online transforms remain, and compare fake-quantized versus exported-model logits under the same quantization settings. The accompanying artifact includes the code, frozen protocol, tests, and launch commands for these checks. **Detailed Hyperparameters for QAD.** For the primary training runs presented in Section 5.1, we execute our global quantization-aware distillation (QAD) on a node with 8 GPUs. The detailed optimization recipe is configured as follows:

- **Microsteps and Batching:** The recipe runs for $2,000$ microsteps with a per-rank (per-GPU) batch size of 2 and a gradient accumulation steps of 4. This setup corresponds to a global batch size of 64 and a total of $32,000$ rank-samples, resulting in exactly 500 optimizer updates.
- **Learning Rate Schedule:** We employ the AdamW (Loshchilov & Hutter, 2019) optimizer with a cosine learning-rate decay schedule and a 0.1 warmup ratio (first 50 updates). The peak learning rate is set to 5×10^{-6} for LLaMA-2-7B, Mistral-7B, and Qwen2.5-7B. For the standard LLaMA-3-8B forward-KL runs, the peak learning rate is set to 1×10^{-5}.
- **Loss Configuration:** The distillation temperature τ in Equation 6 is fixed to 2. The backward pass utilizes the standard identity straight-through estimator (STE) for rounding gradients.

Table 4: Canonical local RTN W4 G128/A4 endpoint results in the nohad graph. PPL is measured on the full WikiText-2 test set. Each Δ_{adapt} compares PTQ and QAD from the same initializer under the same deployment evaluator; it is not the search-space contrast in Table 2.

Model	Initializer	PTQ PPL ↓	QAD PPL ↓	Δ_{adapt}
LLaMA-2-7B	FIXED0	19.4701	8.6489	10.8212
LLaMA-2-7B	learned rotation	17.4621	9.5285	7.9336
LLaMA-3-8B	FIXED0	59.2573	13.7320	45.5253
LLaMA-3-8B	learned rotation	60.3299	13.7372	46.5927

Table 5: Matched post-QAD initialization ablation. All rows use the same QAD budget; only the initialization changes. Values are internal RTN W4A4 perplexities.

Initialization	LLaMA-2	LLaMA-3
Fixed randomized Hadamard	7.98	13.78
Cayley-learned Hadamard initialization	7.97	13.87
Gaussian random orthogonal (QR)	8.05	16.31
Identity after normalization fusion	13.49	19.33
Plain FP coordinates	9.17	15.46

Pretrained and trained model checkpoints and datasets are not redistributed and must be obtained or reproduced under their original access conditions.

C Additional Experimental Results

C.1 Direct Adaptation at Fixed Initializers

Table 4 isolates the effect of direct adaptation while holding the initializer and canonical deployment evaluator fixed. Each PTQ–QAD pair uses the same initializer and nohad graph, so $\Delta_{\text{adapt}} = \text{PPL}_{\text{PTQ}} - \text{PPL}_{\text{QAD}}$ measures the endpoint change after direct weight adaptation. Unlike Table 1, it is not a comparison with a separately optimized SpinQuant-nohad baseline. Unlike Δ_{space}, it does not compare rotation-only and direct-weight optimization under a common training budget. The learned initializer is a diagnostic because its rotation was calibrated separately; FIXED0 is the data-free primary initializer.

Direct weight adaptation lowers canonical RTN perplexity for all four paired endpoints. The near equality of the two LLaMA-3 QAD endpoints indicates that learned rotation is not necessary to obtain the final endpoint in this setting; it does not establish that initialization is irrelevant to optimization efficiency or stability.

C.2 Initialization

Proposition 1 does not specify which initialization a finite optimizer should use. To isolate this optimization question, Table 5 reports a matched internal evaluator used during the initialization sweep. Its absolute perplexity scale differs from the standalone canonical evaluator in Tables 2 and 4; comparisons should be made only within this table.

Three conclusions follow.

Rotation learning is not consistently useful after QAD. The zero-step fixed initializer and the 100-step Cayley-learned initializer reach nearly identical matched PPLs. Independent deployed evaluations likewise show no systematic accuracy benefit from Cayley learning. This does not prove that learned rotations are universally useless; it shows that their PTQ advantage is absorbed by global QAD under the tested setting.

Coordinate dispersion matters. The identity initialization is substantially worse than either orthogonal initializer. Identity here is not identical to the untouched FP model: it passes through the

normalization-fusion preprocessing path while setting the rotation to I. This is the correct internal control for isolating the rotation within that pipeline.

Hadamard structure is robust but not universally necessary. Random QR and randomized Hadamard are nearly tied on LLaMA-2. On LLaMA-3, however, the randomized Hadamard initializer is better by 2.53 perplexity. The evidence therefore supports Hadamard as a robust default, rather than the stronger claim that it is uniquely optimal on every model.

C.3 Longer Direct-FP Training

We extend direct-FP and rotated-initialization runs to $4{,}000$ microsteps. On LLaMA-2, the direct-FP QAD trajectory becomes unstable at approximately step $2{,}700$ and does not recover, whereas the rotated run remains stable. On LLaMA-3, direct-FP QAD does not diverge, but it plateaus at approximately 14.9 internal RTN perplexity, compared with approximately 14.0 for the rotated initialization.

These internal-evaluator trajectories argue against the narrow explanation that the direct-FP control merely requires a few additional steps under the tested schedule. They do not establish that no alternative optimizer, learning-rate schedule, or substantially larger dataset could close the gap. They are optimization diagnostics rather than a comparison of the feasible-set infima in Proposition 1.

C.4 Lower Weight Precision

Table 6: Internal-evaluator bit-width sweep. Activations remain four-bit and both initializations use matched global QAD. NaN denotes a failed computation.

	LLaMA-2		LLaMA-3	
Bits	**Rotated**	**FP init**	**Rotated**	**FP init**
W4A4	7.97	9.17	13.87	15.46
W3A4	10.37	11.68	18.07	19.67
W2A4	1082	5464	3929	4474
W1A4	560	NaN	4434	6165

The rotated initialization is lower in all seven finite pairs. W1A4 and W2A4 collapse for both initializations under the fixed $2{,}000$-microstep budget, so these internal results are optimization diagnostics rather than deployable model-quality evidence. The nonmonotonicity among these poor runs is not evidence of a one-bit advantage.

C.5 Retaining Online Hadamard Transformations

With the FIXED0 initializer and online Hadamard enabled, matched RTN W4A4 QAD improves LLaMA-2 PPL from 9.1120 to 6.5354, and LLaMA-3 from 11.2831 to 9.5081. This shows that weight adaptation can help within another fixed deployment graph; it does not make runtime and offline training costs interchangeable. The strongest local SpinQuant-had GPTQ references remain lower, but use a different quantizer. We make no latency claim because these experiments include no traceable device-and-workload benchmark.

D Baseline Provenance

This section separates published SpinQuant references from local reproductions. GPTQ uses approximate second-order weight reconstruction (Frantar et al., 2023); RTN rounds directly, and their scores are not interchangeable.

D.1 SpinQuant reproduction record

The published values were cross-checked against the W4A4KV16 rows of Table 1 (GPTQ) and Table 16 (RTN) in Liu et al. (2025), arXiv version 4. Here KV16 denotes 16-bit attention key/value

Table 7: Source-separated SpinQuant W4A4 WikiText-2 perplexity provenance, with GPTQ/RTN in each cell. Published W4A4KV16 references (Liu et al., 2025), best local, unswept 100-step (s100), and author-released-rotation results remain separate; local cache precision is unspecified. GPTQ and RTN entries use distinct deployment quantizers and are not a matched ranking. N/R means not reported in the cited paper; N/E means no verified measurement. Lower is better within a fixed protocol.

Model	Deployment	Published GPTQ/RTN	Best local GPTQ/RTN	Local s100 GPTQ/RTN	Our eval. with released rotations GPTQ/RTN
LLaMA-2-7B	had	5.9 / 6.1	5.99 / 10.64	6.97 / N/E	N/E / N/E
LLaMA-2-7B	nohad	9.2 / N/R	13.45 / 13.52	22.36 / 13.52	N/E / N/E
LLaMA-3-8B	had	7.1 / 7.7	7.56 / N/E	8.78 / 9.81	8.77 / N/E
LLaMA-3-8B	nohad	18.6 / N/R	39.73 / 30.85	118.86 / 30.85	58.32 / N/E

caches. The local cache precision is not recorded, so the published column is a reference rather than a fully matched comparison. The other columns are the study's measurements. Author-released rotations are learned matrices, not complete pre-quantized models. They were applied to the base model and evaluated through the study's own quantization pipeline. Thus the released-rotation measurements are not results reported by the SpinQuant authors.

The LLaMA-3 released rotations obtain 8.77 with online transforms and 58.32 without them under GPTQ. They therefore do not close the nohad reproduction gap: the best local nohad GPTQ result is 39.73, versus the published 18.6. The LLaMA-2 best local nohad GPTQ result is 13.45, versus the published 9.2. No causal explanation for these gaps is established here.

The unswept s100 column is not an evaluation of released rotations and is not interchangeable with the best-local sweep. For example, its LLaMA-3 nohad GPTQ result is 118.86, and its had RTN result is 9.81. The nohad RTN values 13.52 and 30.85 are historical legacy-affine measurements, not canonical primary baselines; no value in this table is used for the canonical RTN deltas in the main text. Similarly, the LLaMA-2 best-local had RTN result of 10.64 is retained only as source-separated provenance even though it is not close to the published 6.1.

E Exploratory Diagnostics and Their Interpretation

E.1 Interpolation between independently trained endpoints

For endpoints $u_A, u_B \in \mathbb{R}^p$ expressed in a common parameter layout, define $u(\alpha) = (1-\alpha)u_A + \alpha u_B$, $\alpha \in [0,1]$. A diagnostic must specify the common graph, the quantization rule along the path, the evaluated loss, and whether endpoints have been aligned with respect to parameter symmetries. For a finite set $\mathcal{S} \subset [0,1]$ containing $0, 1$ and a specified scalar evaluation loss ℓ that is finite on the sampled path, one possible *chord-relative* barrier is

$$B_{\mathcal{S}}(u_A, u_B) = \max_{\alpha \in \mathcal{S}} \left[\ell(u(\alpha)) - (1-\alpha)\ell(u_A) - \alpha\ell(u_B)\right]. \tag{32}$$

The diagnostic code records this as the linear-reference barrier and also reports the simple barrier

$$B_{\mathcal{S}}^{\text{simple}}(u_A, u_B) = \max_{\alpha \in \mathcal{S}} \ell(u(\alpha)) - \max\{\ell(u_A), \ell(u_B)\}. \tag{33}$$

Using 11 uniformly spaced interpolation points and quantized loss, the simple barriers are 7.96 for the LLaMA-2 FP–fixed path, 7.46 for its nohad–fixed control, and 10.26 for the LLaMA-3 FP–fixed path. The artifact records both loss curves and both barrier definitions. A positive barrier for the control weakens any claim of a unique FP-to-rotation obstruction. Independently trained solutions can differ by coordinate symmetries and optimization histories, and no endpoint symmetry alignment was performed. These values are therefore exploratory observations rather than proof that a particular initializer removes an unavoidable optimization barrier.

E.2 EXCESS KURTOSIS: DEFINITION AND PROVENANCE

For activation channel c with $n \geq 2$ observations $x_{1,c}, \ldots, x_{n,c} \in \mathbb{R}$, define

$$\bar{x}_c = \frac{1}{n}\sum_{i=1}^{n} x_{i,c}, \qquad m_{2,c} = \frac{1}{n}\sum_{i=1}^{n}(x_{i,c} - \bar{x}_c)^2, \tag{34}$$

$$m_{4,c} = \frac{1}{n}\sum_{i=1}^{n}(x_{i,c} - \bar{x}_c)^4, \qquad \kappa_c = \frac{m_{4,c}}{m_{2,c}^2} - 3 \quad (m_{2,c} > 0). \tag{35}$$

This is the moment-ratio definition, not an unbiased finite-sample estimator. The fourth power in $m_{4,c}$ is essential. The diagnostic uses float64 streaming central moments over eight WikiText-2 windows for each input to 224 linear modules, excluding the language-modeling head. It computes pooled excess kurtosis per module, discards nonfinite values, and then reports means, maxima, medians, and layer/type aggregates. Thus the observations below are reproducible from the accompanying code and result files, but remain small-sample diagnostics.

The median per-module excess kurtosis is 15.15 for direct-FP LLaMA-2 and 11.40 for direct-FP LLaMA-3. Fixed and learned rotated solutions have medians approximately zero. A few modules, particularly some down projections, have large values that dominate the mean.

For each model the record describes a correlation across three pairs (k_j, p_j), one per initialization, where k_j is its reported mean kurtosis and p_j its deployed PPL. With $\bar{k} = \frac{1}{3}\sum_{j=1}^{3} k_j$ and $\bar{p} = \frac{1}{3}\sum_{j=1}^{3} p_j$, Pearson's correlation is

$$r = \frac{\sum_{j=1}^{3}(k_j - \bar{k})(p_j - \bar{p})}{\sqrt{\sum_{j=1}^{3}(k_j - \bar{k})^2 \sum_{j=1}^{3}(p_j - \bar{p})^2}}, \tag{36}$$

provided both sums of squares are positive. Recalculation from the recorded triples gives $r = 0.997$ for LLaMA-2 and $r = 0.979$ for LLaMA-3. The associated PPL values use a historical per-output-channel evaluator rather than the canonical G128 protocol. Three points do not establish a robust predictive relationship or causality, and kurtosis does not uniformly track downstream accuracy in this diagnostic.