BEYOND ROTATIONS: AUROOFT FOR EXPRESSIVE QUANTIZED ORTHOGONAL FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Quantized orthogonal fine-tuning (QOFT) adapts low-bit language models by learning structured activation rotations before frozen quantized weights, but its task-specific updates at each adapted layer remain constrained to a linear orthogonal family. We treat this as a layer-local design hypothesis rather than a demonstrated failure mode: rotations alone may be too narrow for the input-dependent corrections that mathematical reasoning requires. We introduce AuroOFT, which keeps QOFT as a stable quantization-compatible branch and attaches a zero-start gated low-rank nonlinear residual to each adapted linear layer. The residual maps activations into an RMS-normalized compact latent space and applies adaptive nonlinear bases with bounded or token-dependent gating. The zero-initialized up projection makes AuroOFT functionally identical to QOFT at initialization, while orthogonality remains a branch-level property rather than a property of the combined layer. Under matched data, decoding, and parser protocols on six mathematical-reasoning benchmarks, AuroOFT improves Macro-6 over matched QOFT by 0.38–2.70 points on the four 1.5B/3B Qwen2.5 settings, and it trails QOFT at 7B. We report this pattern as scale- and task-dependent evidence. Parameter-matched linear controls attribute the 3B increment mainly to the nonlinear bases, while a linear control matches AuroOFT on the instruct settings. Much of the advantage over QLoRA (6.52–10.62 points at 1.5B/3B) is already provided by the QOFT carrier that AuroOFT contains at initialization. On top of this carrier, the residual adds a further increment while saving 32.3–44.7% of the trainable parameters that QLoRA requires. The small exam-style multiple-choice math set is treated only as a protocol-sensitivity diagnostic, and all headline comparisons are single runs under a leakage-audited selection protocol, with multi-seed and unified-script replications of the 3B pair giving consistent increments. Our code is available at the anonymous repository `https://anonymous.4open.science/r/AuroOFT-F3FD`.

1 INTRODUCTION

Parameter-efficient fine-tuning has become the standard route for adapting large models under realistic compute and storage budgets (Lialin et al., 2023). For low-bit reasoning models, however, an effective adapter should behave like a ***small but faithful correction layer***. It must fit beside a frozen quantized foundation model, preserve useful pretrained geometry, and still express the input-dependent corrections required by mathematical reasoning. Figure 1 summarizes this tension. Additive adapters can be flexible but need not preserve geometry, while orthogonal adapters are stable but may restrict the local functions available for adaptation. The central question is therefore not only whether an adapter can be made small, but what kind of correction a small adapter is allowed to express.

This issue is especially visible in mathematical reasoning. A model may preserve broad pretrained knowledge and still fail because a local symbolic step, choice elimination, or numerical comparison requires an input-dependent correction. LoRA and QLoRA obtain efficiency by constraining updates to compact low-rank linear subspaces (Hu et al., 2022; Dettmers et al., 2023). OFT, OFTv2, and QOFT instead use structure-preserving orthogonal transformations, with QOFT applying an input-side Cayley-Neumann rotation before frozen NF4 weights (Qiu et al., 2023; 2025). This gives a

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.

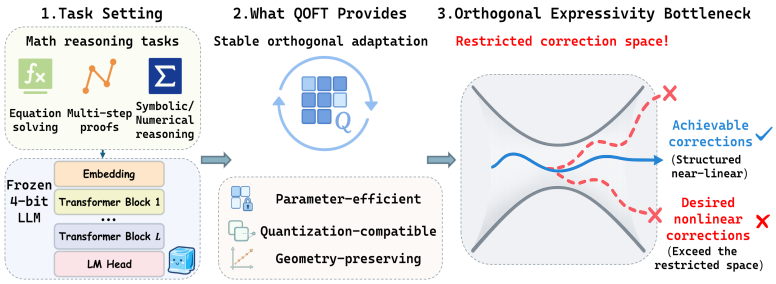

Figure 1: Research problem and challenge. Low-bit reasoning adapters must balance quantized stability with expressive correction capacity.

stable quantized path, but the task-specific change at each adapted layer remains an orthogonal linear transformation. Nonlinear low-rank methods such as LoRAN and AuroRA show that compact adapters need not remain linear (Li et al., 2024a; Dong et al., 2025), yet they are designed around LoRA-style additive updates rather than the geometric backbone that motivates QOFT.

We formulate this observation as a ***layer-local design hypothesis***, which we call the ***orthogonal expressivity bottleneck***. The hypothesis is deliberately scoped. The full Transformer that contains QOFT is of course nonlinear. What QOFT restricts is the task-specific update at each adapted layer, which must belong to an input-side orthogonal linear family. For reasoning, corrections may depend on the current token state, the latent arithmetic context, or a nonlinear interaction that is hard to represent by rotation alone, so this layer-local family can be too narrow. We use the bottleneck as a design hypothesis and test its consequences empirically rather than as a demonstrated cause of reasoning failures. Simply inserting a nonlinear module into the rotation path would blur the geometric meaning of QOFT, while replacing QOFT would discard the stability that made it useful.

Can we keep QOFT's quantized orthogonal path as the stable backbone while giving each layer expressive nonlinear corrections? Our answer is AuroOFT, a zero-start nonlinear residual augmentation for quantized orthogonal fine-tuning. Each adapted layer keeps the QOFT branch unchanged. Activations are rotated by the learned Cayley-Neumann adapter and then passed through the frozen NF4 weight. In parallel, a lower branch applies a down projection, an ANL, a zero-initialized up projection, and a gate. The branches meet by residual addition (Figure 2). The construction applies layer-wise: any linear projection (attention query/key/value or MLP up/down) can be wrapped, either across all linear layers or on selective targets.

The defining principle is ***orthogonality as a stable quantized backbone, nonlinearity as a zero-start residual correction***. The nonlinear branch expands the layer-wise function family without modifying the quantized base weight or inserting nonlinear operations inside the Cayley-Neumann rotation. Zero initialization of the up projection makes the initial model exactly function-equivalent to QOFT, while the input-dependent residual deliberately turns the combined layer into a nonlinear mapping. Thus, only the QOFT branch preserves orthogonal structure. AuroOFT borrows the nonlinear-low-rank principle behind AuroRA, but changes the augmented object: the nonlinear map is placed beside a quantized orthogonal QOFT layer, not in place of it.

This distinction also shapes how the method should be evaluated. The primary comparison is against matched QOFT: it isolates whether the nonlinear residual supplies useful expressivity beyond the stable orthogonal carrier. The comparison with QLoRA is secondary and mainly tests parameter efficiency. Because QOFT already closes most of the gap to QLoRA at the evaluated scales, we report the decomposition explicitly rather than letting the aggregate comparison carry the claim. All comparisons use fixed data, decoding, and parser conditions, and treat small or protocol-sensitive benchmarks as diagnostics rather than standalone evidence.

This paper contributes a compact framework for turning the above hypothesis into a testable adaptation object.

- **A scoped design hypothesis.** The ***orthogonal expressivity bottleneck*** states that QOFT's input-side rotations restrict each adapted layer's task update to a structured linear family. It is a layer-local hypothesis to be tested, not a demonstrated cause of reasoning failures.

- **An adaptation object.** AuroOFT instantiates the hypothesis as a zero-start gated nonlinear low-rank residual branch parallel to QOFT, with tanh, normalized spline, and enhanced dual-ANL variants, applicable layer-wise to all or selected linear projections.
- **A boundary analysis.** The formulation makes explicit what is preserved and what is not: exact initialization equivalence to QOFT, containment of the QOFT family, branch-limited orthogonality, and a non-mergeable residual cost during inference.
- **A scoped evaluation.** Mathematical reasoning experiments align data, decoding, and parsers, with the increment over matched QOFT as the primary measure, QLoRA as a secondary parameter-efficiency reference, and implementation-level caveats disclosed in the appendix.

2 Related Work

Stable and quantized adaptation. PEFT methods differ in the adaptation object they choose. These include bottleneck modules (Houlsby et al., 2019), prompts (Li & Liang, 2021; Liu et al., 2022b), activation scalings (Liu et al., 2022a), and low-rank additive matrices (Hu et al., 2022). Quantized adaptation makes this choice sharper. GPTQ, SmoothQuant, and AWQ reduce the frozen model footprint (Frantar et al., 2023; Xiao et al., 2023; Lin et al., 2024), while QLoRA, LoftQ, and QA-LoRA attach LoRA-style updates to low-bit weights (Dettmers et al., 2023; Li et al., 2024b; Xu et al., 2024).

Orthogonal PEFT changes the prior by learning geometry-preserving rotations. OFT introduces orthogonal fine-tuning (Qiu et al., 2023), BOFT compresses rotations with butterfly factors (Liu et al., 2024b), and OFTv2/QOFT scales the idea through input-centric Cayley-Neumann rotations before frozen quantized weights (Qiu et al., 2025). HRA explicitly bridges the two paradigms by composing low-rank updates with products of Householder reflections (Yuan et al., 2024), confirming that the orthogonal-versus-low-rank axis is a central design dimension. These lines emphasize efficiency, low-bit compatibility, and stability, but still leave a common gap for reasoning: the task correction is a linear additive map or an input-side (approximately) orthogonal rotation. AuroOFT keeps the quantized orthogonal carrier while adding a separate nonlinear correction object, and its residual branch is agnostic to whether the carrier is parameterized by Cayley-Neumann rotations, butterfly factors, or Householder products.

Expressive residual adaptation. Expressive PEFT asks what a compact adapter can represent, not only how many parameters it adds. LoRA analyses and variants such as AdaLoRA, DyLoRA, DoRA, VeRA, FourierFT, MiLoRA, MoSLoRA, and PeriodicLoRA enrich low-rank updates through rank allocation, shared bases, magnitude-direction decomposition, spectral coefficients, mixed subspaces, or staged rank growth (Zeng & Lee, 2023; Zhang et al., 2023; Valipour et al., 2023; Liu et al., 2024a; Kopiczko et al., 2024; Gao et al., 2024; Wang et al., 2025; Wu et al., 2024; Meng et al., 2024). LoRAN and AuroRA further insert nonlinear hidden mappings, including tanh and B-spline components, into the low-rank path (Li et al., 2024a; Dong et al., 2025). These works show that compact adaptation need not be purely linear, but they are designed around LoRA-style additive updates rather than a quantized orthogonal main branch.

AuroOFT uses their expressivity lesson in a different role. The nonlinear map is zero-started, gated, and residual beside QOFT, so ablations can vary only the correction family while the orthogonal carrier remains fixed. This distinction is important for attribution. The proposed residual is not an alternative backbone, but a controlled way to test whether nonlinear corrections can expand QOFT without changing its quantized orthogonal carrier. The comparison therefore follows a narrow principle. Preserve the stable branch, vary the residual branch, and measure whether the extra expressivity improves adaptation under matched conditions. This principle bridges prior adapter design to the AuroOFT layer formalized next.

3 Preliminaries

QOFT. Let $x \in \mathbb{R}^{d_{\text{in}}}$ be the input activation of a linear layer with frozen quantized weight $W_{q,l}$, stored in $\mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$ following the implementation's weight-storage convention, so that

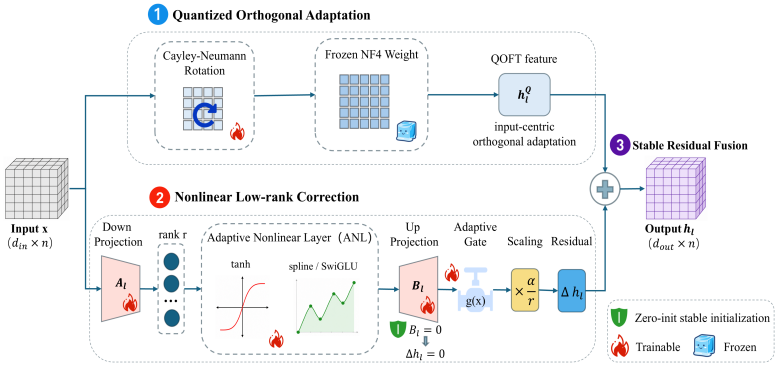

Figure 2: AuroOFT layer. A stable QOFT branch is augmented by a zero-start gated nonlinear low-rank residual.

$\mathrm{Dequant}(W_{q,l})^\top \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$. QOFT applies an orthogonal transformation to the activation side before the quantized linear map.

$$h_l^Q = \mathrm{Dequant}(W_{q,l})^\top R_l^\top x, \tag{1}$$

where $R_l \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}}$ is the learned rotation acting on the activation, parameterized by a skew-symmetric matrix through a Cayley transform. OFTv2 replaces the inverse in the Cayley transform with a truncated Neumann series, so the realized rotations are approximately rather than exactly orthogonal. The important point for AuroOFT is that the orthogonal adapter acts before the frozen NF4 weight. The quantized base weight is not rotated or modified.

AuroRA-style nonlinear low-rank mapping. LoRA uses a two-projector linear update.

$$h = W_0 x + BAx, \tag{2}$$

which is still linear in x for fixed A and B. AuroRA inserts an adaptive nonlinear layer $\sigma(\cdot)$ between the two projectors.

$$h = W_0 x + B\sigma(Ax). \tag{3}$$

Its ANL combines fixed nonlinear activations and learnable B-spline components. AuroOFT adopts this nonlinear-low-rank principle as a residual beside QOFT rather than replacing a LoRA branch.

4 Method

4.1 AuroOFT Layer

Figure 2 shows the layer-level construction, and Algorithm 1 summarizes the corresponding forward computation. AuroOFT wraps an already adapted OFT/QOFT linear module rather than replacing it. The upper branch is the stable quantized orthogonal path inherited from QOFT. The lower branch is a nonlinear low-rank residual evaluated on the same activation and added only at the output.

For an adapted layer l, AuroOFT computes the following branch input and residual output.

$$z_l = A_l x, \qquad \tilde{z}_l = \mathcal{N}_l(z_l), \tag{4}$$

$$\begin{aligned} h_l &= h_l^Q + \Delta h_l, \\ \Delta h_l &= G_l(\tilde{z}_l)\frac{\alpha}{r} B_l \,\mathrm{Drop}(\phi_l(\tilde{z}_l)), \end{aligned} \tag{5}$$

where h_l^Q is the QOFT output in Eq. 1, $A_l \in \mathbb{R}^{r \times d_{\text{in}}}$ is the down projection, $B_l \in \mathbb{R}^{d_{\text{out}} \times r}$ is the up projection, $\mathcal{N}_l$ is either the identity map or FP32 RMS normalization, ϕ_l is an adaptive nonlinear layer in the compact rank-r space, α/r is the residual scaling factor, and G_l is a scalar, bounded scalar, or token-dependent gate.

This parallel placement is the defining design choice. The nonlinear branch is not inside the Cayley-Neumann rotation and does not change the orthogonal parameterization. It provides an additional input-dependent correction after the QOFT feature has been computed, and the zero-initialized up projection ensures that this additional path starts inactive.

Algorithm 1 AuroOFT Layer Forward Pass

Input. activation x, wrapped QOFT layer f_l^Q, branch modules $\{A_l, \mathcal{N}_l, \phi_l, B_l, G_l\}$
Output. adapted feature h_l
$h_l^Q \leftarrow f_l^Q(x)$ *stable QOFT branch*
$z_l \leftarrow A_l x$ *down projection*
$\tilde{z}_l \leftarrow \mathcal{N}_l(z_l)$ *identity or FP32 RMSNorm*
$u_l \leftarrow \mathrm{Drop}(\phi_l(\tilde{z}_l))$ *ANL in rank-r space*
$g_l \leftarrow G_l(\tilde{z}_l)$ *scalar, bounded, or token gate*
$\Delta h_l \leftarrow g_l \frac{\alpha}{r} B_l u_l;\ h_l \leftarrow h_l^Q + \Delta h_l$
return h_l
Zero-start. Initialize $B_l = 0 \Rightarrow \Delta h_l = 0$, so the layer starts exactly as QOFT.

4.2 LITE ANL

The lightweight variant uses a two-tanh hidden transformation plus a learnable per-dimension residual scale.

$$\phi_l^{\mathrm{Lite}}(z) = \tanh(H_l \tanh(z)) + s_l \odot \tanh(z), \tag{6}$$

where $z = \tilde{z}_l$, $H_l \in \mathbb{R}^{r\times r}$, and $s_l \in \mathbb{R}^r$. The code initializes H_l as the identity and s_l to a small positive value. This variant is computationally simple and isolates whether nonlinear hidden mixing alone is useful beyond QOFT.

4.3 SPLINENORM ANL

The normalized spline variant follows the AuroRA motivation more closely while adapting it to the AuroOFT branch.

$$\bar{z} = \tanh(\mathrm{RMSNorm}_{\mathrm{FP32}}(z)), \tag{7}$$

$$\phi_l^{\mathrm{SplineNorm}}(z) = \tanh(H_l \tanh(\bar{z})) + C_l \Psi(\bar{z}), \tag{8}$$

where $\mathcal{B}_j(\cdot)$ denotes the j-th B-spline basis response on the bounded interval $[-1, 1]$, $\Psi_j(\bar{z}) = \sum_{i=1}^{r} \mathcal{B}_j(\bar{z}_i)$ is the implementation's rank-dimension aggregated spline feature, and $C_l \in \mathbb{R}^{r\times(M+p)}$ is the learnable spline coefficient matrix for grid size M and order p. The FP32 RMS normalization is applied once to the branch input z_l. The two tanh maps above then operate on the already-normalized, bounded variable $\bar{z}$ with no further normalization. The FP32 RMS normalization and tanh bounding are used to keep low-dimensional activations within the spline grid. This is an adaptation of the AuroRA ANL idea, not a direct transplantation of its LoRA theory.

4.4 ENHANCED ANL AND GATING

The enhanced branch uses an FP32-normalized compact representation

$$z = \mathrm{RMSNorm}_{\mathrm{FP32}}(A_l x), \tag{9}$$

where the normalization is applied once to the branch input, and combines two nonlinear bases.

$$u_t = \tanh(P_{t,l} \tanh z), \tag{10}$$

$$u_g = \mathrm{SiLU}(P_{g,l} z) \odot \tanh(P_{v,l} z). \tag{11}$$

For the dual ANL, the final hidden output is

$$\phi_l^{\mathrm{Dual}}(z) = \pi_{t,l} u_t + \pi_{g,l} u_g, \tag{12}$$

$$[\pi_{t,l}, \pi_{g,l}] = \mathrm{softmax}(m_l). \tag{13}$$

The code supports scalar, bounded, and token-dependent gates.

$$G_l(z) = \eta_l, \tag{14}$$

$$G_l(z) = 2\sigma(\gamma_l), \tag{15}$$

$$G_l(z) = 2\sigma(W_{g,l} z), \tag{16}$$

corresponding respectively to the unconstrained scalar, bounded scalar, and token-dependent gate. In the enhanced implementation, η_l is initialized to one, while $\gamma_l = 0$ and $W_{g,l} = 0$ make the bounded and token gates initialize to one. This avoids the small-gate start that can suppress early gradients through the zero-initialized up projection.

4.5 Layer Injection and Optimization

AuroOFT supports all-layer injection or selective targets such as query/value and query/key/value plus MLP up/down projections. The QOFT branch and nonlinear residual use separate parameter groups, with branch-specific learning rate and gradient clipping. In the released scripts, parameters are assigned to these groups through name-based matching rules. We disclose a known subtlety of this implementation: substring rules can also match unrelated module paths such as Qwen's `gate_proj` projections, which may route a small subset of QOFT parameters into the residual-branch group. Together with other implementation-level differences between the released QOFT and AuroOFT scripts, this is documented in Appendix B and treated as a limitation of the current matched protocol. Disabling AuroOFT recovers the same QOFT training interface.

4.6 Structural Properties

Proposition 1 (Zero-start equivalence)**.** *If $B_l = 0$ for every nonlinear branch, then the initialized AuroOFT model computes exactly the same function as the corresponding QOFT model on every input sequence.*

Proof. For each adapted layer, $B_l = 0$ implies $\Delta h_l = 0$ in Eq. 5, regardless of A_l, ϕ_l, dropout, or gate values. Hence $h_l = h_l^Q$ at every adapted layer, and the layer-wise equality composes through the network.

Proposition 2 (QOFT containment)**.** *The function family induced by AuroOFT contains the corresponding QOFT family when the nonlinear residual is inactive.*

Proof. Any QOFT model can be represented by setting the nonlinear residual to zero, for example through $B_l = 0$ at all adapted layers. Thus QOFT is a subfamily of AuroOFT. We use the up-projection condition because it holds for all implemented gate types, including bounded and token-dependent gates whose finite-parameter values are positive.

Remark 1 (What the containment does not prove)**.** *Containment is only an expressivity statement. Under ideal global optimization, the best empirical objective is no worse than QOFT. It does not guarantee finite-step accuracy gains or justify test-set-based selection.*

Orthogonality boundary. The main branch retains QOFT's approximate orthogonal structure, but the full AuroOFT layer adds an input-dependent nonlinear residual and is not an orthogonal transformation, so we claim no global angle, norm, or forgetting guarantees for the combined mapping.

Non-mergeability and cost. Because Δh_l is nonlinear in the current activation, the residual cannot generally be merged into a static post-training weight matrix. Unlike linear LoRA-style adapters, AuroOFT therefore adds inference cost from projection, nonlinear computation, gating, and up projection.

Parameter cost. Each adapted layer adds trainable parameters beyond the QOFT branch in that layer, with the amount depending on the variant.

$$\text{Lite:} \quad rd_{\text{in}} + r^2 + rd_{\text{out}} + r + \kappa_g, \tag{17}$$

$$\text{SplineNorm:} \quad rd_{\text{in}} + r^2 + rd_{\text{out}} + r(M+p) + \kappa_g, \tag{18}$$

$$\text{Enhanced dual-ANL:} \quad rd_{\text{in}} + rd_{\text{out}} + 3r^2 + 2 + \kappa_g, \tag{19}$$

where the rd_{in} and rd_{out} terms correspond to A_l and B_l, the middle terms correspond to the variant-specific hidden map (H_l, the spline coefficients with $M+p$ basis responses, or the three r^2 projections for tanh, SwiGLU gate, and SwiGLU value), the r term is the Lite residual scale s_l, the two additional constants are the dual mixing logits, and κ_g is 1 for scalar or bounded scalar gating and r for token-dependent gating. In all variants, the dominant additional computation scales as $O(rd_{\text{in}}+r^2+rd_{\text{out}})$ per token and adapted layer, with a small spline-basis overhead for SplineNorm.

Model	Type	Params	Macro-6	AMC23	AQUA	CMATH	GaoKao	Minerva	Olymp.	SAT
1.5B-it	Baseline	-	31.65	17.50	49.20	65.20	36.40	9.60	12.00	59.40
	QLoRA	18.46M	27.60	15.00	42.50	61.50	29.60	8.10	8.90	59.40
	QOFT	7.89M	36.05	27.50	53.10	68.50	41.00	11.80	14.40	81.20
	AuroOFT	10.20M	37.35	27.50	55.70	72.20	41.30	11.80	15.60	87.70
1.5B	Baseline	-	5.35	0.00	18.90	4.00	4.20	2.60	2.40	28.10
	QLoRA	18.46M	26.45	15.00	37.40	64.20	26.80	8.50	6.80	62.50
	QOFT	7.89M	31.53	22.50	53.10	56.30	36.10	8.50	12.70	87.50
	AuroOFT	10.20M	33.37	23.50	46.70	69.40	38.70	8.80	13.10	87.50
3B-it	Baseline	-	30.98	20.30	34.20	42.50	55.60	13.20	20.10	59.60
	QLoRA	24.73M	37.22	24.30	30.50	85.90	52.70	14.60	15.30	66.20
	QOFT	13.00M	43.35	45.00	31.10	84.30	52.50	20.20	27.00	71.90
	AuroOFT	16.74M	43.73	35.00	35.00	85.30	56.60	23.20	27.30	65.60
3B	Baseline	-	23.07	17.60	29.80	39.60	28.30	5.60	17.50	48.20
	QLoRA	24.73M	34.63	27.40	28.50	77.10	48.80	12.40	13.60	61.30
	QOFT	13.00M	42.55	42.50	30.40	85.20	50.30	20.10	26.80	68.50
	AuroOFT	16.74M	45.25	43.30	34.20	87.60	54.60	22.10	29.70	68.60
7B-it	Baseline	-	47.95	50.00	16.50	89.30	61.80	33.50	36.60	53.10
	QLoRA	40.37M	43.67	30.00	48.00	88.80	50.10	25.40	19.70	68.80
	QOFT	17.55M	58.10	52.50	70.90	90.50	63.60	33.50	37.60	96.90
	AuroOFT	22.60M	52.47	55.60	46.80	88.20	58.10	26.30	39.80	96.80
7B	Baseline	-	37.65	25.00	55.10	61.20	42.90	11.80	29.90	71.90
	QLoRA	40.37M	40.78	35.00	48.80	73.70	49.90	18.80	18.50	62.50
	QOFT	17.55M	50.77	52.50	59.40	80.70	55.60	21.70	34.70	87.50
	AuroOFT	22.60M	50.43	53.70	44.30	86.90	56.80	24.70	36.20	92.00

Table 1: Pass@1 mathematical reasoning results across Qwen2.5 model scales. Macro-6 averages the six non-SAT columns and is computed from unrounded per-benchmark scores. Model names are abbreviated for width, and the suffix -it denotes the instruct variant. Reported AuroOFT rows use the variant selected by the proxy-screening protocol for each scale (Appendix B), and the 3B ablation (Table 2) reports the Enhanced-r2 setting.

Variant	Branch Setting	Params	Macro-6	AMC23	AQUA	CMATH	GaoKao	Minerva	Olymp.	SAT
Matched QOFT	no nonlinear branch	13.00M	42.55	42.50	30.40	85.20	50.30	20.10	26.80	68.50
AuroOFT-Lite	tanh, scalar gate, all layers	16.74M	43.42	43.00	31.30	86.00	51.70	20.80	27.70	68.70
AuroOFT-SplineNorm	RMS-bounded B-spline ANL	16.76M	43.78	43.20	31.80	86.40	52.00	21.20	28.10	68.40
AuroOFT-Enhanced-r2	dual ANL, bounded/token gate	16.74M	45.25	43.30	34.20	87.60	54.60	22.10	29.70	68.60
AuroOFT-Enhanced-r4	dual ANL, bounded/token gate	20.49M	45.58	43.80	34.60	87.90	54.90	22.40	29.90	68.30

Table 2: AuroOFT ablation results on Qwen2.5-3B (base). Only AuroOFT-specific residual-branch components are varied, and Macro-6 is computed from unrounded per-benchmark scores.

5 Experiments

5.1 Experimental Setup

Evaluation Principle. The key question is whether the nonlinear residual improves over QOFT under matched adaptation conditions. Unless otherwise stated, methods share the same model, data, sample order, sequence length, quantization, adapter placement, decoding, and parser, and use the same optimizer family and schedule. Implementation-level details of the released scripts are not fully unified between the QOFT and AuroOFT entries, and these differences are disclosed in Appendix B and treated as a limitation when differences are attributed to the residual branch alone.

Training Setup. We use Qwen2.5 models with NF4, OpenR1-Math data, effective batch size 32, OFT block size 32, five Cayley-Neumann terms, and length 16,384 when feasible. A matched manifest fixes shared QOFT fields. Only residual-branch variables vary.

Benchmarks and Metrics. We evaluate AMC23, AQUA, CMATH, GaoKao, Minerva Math, OlympiadBench, and SAT Math. Macro-6 averages the six non-SAT benchmarks and serves as the main aggregate. SAT Math is retained as a protocol-sensitive diagnostic. We report official accuracy plus sample counts, formatting failures, timeouts, parameters, memory, time, and throughput.

Selection Protocol. Model selection uses a leakage-audited proxy set drawn from OpenR1 samples outside the 50k training subset and deduplicated against final tests by normalized text hashes. Test benchmarks do not choose rank, gates, ANL type, branch learning rate, or targets. After freezing a configuration, matched QOFT and AuroOFT runs use the same seed policy, decoding setup, and parser.

5.2 Main Experimental Results

Table 1 reports Pass@1 and Macro-6 across Qwen2.5 scales for the frozen baseline, QLoRA, QOFT, and AuroOFT under the same benchmark columns. All rows are single runs under the selection protocol above. Seed replication with aggregate statistics for the two focal 3B settings is reported in Appendix D (Section 7).

AuroOFT shows its clearest gains over matched QOFT in the 1.5B and 3B settings. Macro-6 improves by 1.30, 1.83, 0.38, and 2.70 points on 1.5B-it, 1.5B, 3B-it, and 3B respectively, with non-degraded metrics on 7/7, 6/7, 5/7, and 7/7 benchmark columns. At 7B the evidence is negative or mixed: AuroOFT trails QOFT by 5.63 Macro-6 points on 7B-it, including a 24.1-point AQUA drop, and by 0.33 points on 7B, while still improving AMC23 and OlympiadBench.

The QLoRA comparison should be interpreted with an explicit decomposition. At 1.5B/3B, AuroOFT leads QLoRA by 6.52–10.62 Macro-6 points, but the QOFT carrier already accounts for 5.08–8.45 of those points (73–94%). The nonlinear residual contributes the remaining 0.38–2.70 points on top of the carrier. We accordingly treat the increment over matched QOFT as the primary measure of the residual branch, and the QLoRA margin as a parameter-efficiency reference. AuroOFT uses fewer trainable parameters than QLoRA at every scale in Table 1, and QLoRA itself trails the frozen baseline on 1.5B-it and 7B-it Macro-6.

5.3 Further Analysis

(I) Ablation Study. Table 2 isolates AuroOFT-specific choices while fixing the base model, quantization, QOFT branch, data, optimizer, decoding, and parser. The rows compare no branch, Lite tanh, bounded SplineNorm, and enhanced dual tanh/SwiGLU variants. Rank/gate rows test added residual capacity under the same evaluation protocol.

(II) Parameter Efficiency Analysis. AuroOFT also uses adapter parameters more efficiently than QLoRA. Table 14 (Appendix D) shows savings of 8.26M, 7.99M, and 17.77M parameters on 1.5B-it, 3B-it, and 7B-it, or 44.7%, 32.3%, and 44.0%. The saved budget can support more task-specialized adapters while keeping correction capacity compact.

(III) Training Dynamics Analysis. Figure 5 (Appendix D) checks optimization stability. AuroOFT starts from the same loss as QOFT and follows a comparable matched-training trajectory. The 120-step variants are diagnostics, not accuracy evidence.

(IV) Parameter-Matched Linear Residual Control. The increment over QOFT in Table 1 does not by itself separate nonlinearity from added capacity (Section 7). Table 3 therefore reports two parameter-matched linear controls on the 3B ablation setting: an *identity-ANL* control that keeps the dual-ANL structure, budget, optimizer group, and injection pattern but replaces the nonlinear bases with the identity map, and a *plain low-rank* branch that is rank-matched to the same budget. The identity-ANL row is complete, the low-rank row reports its Macro-6 value, and the remaining – cells await the corresponding runs with no values imputed. The two controls recover only 0.55 and 0.65 of the 2.70-point increment, and cross-scale controls in Appendix D make the attribution scale-dependent.

6 Discussion

AuroOFT treats orthogonality as a stable low-bit backbone, not a global constraint, and rank-r bottlenecks, α/r scaling, and bounded or token gates keep the residual a correction. The residual itself is therefore not isometric, and this is a deliberate trade in which the layer buys input-dependent expressivity and recovers control through zero-start and gating rather than through orthogonality.

Variant	Branch Setting	Params	Macro-6	AQUA	CMATH
Matched QOFT	no nonlinear branch	13.00M	42.55	30.40	85.20
Linear control (identity-ANL)	identity map, dual-ANL budget	16.74M	43.2	35.1	88.2
Linear control (low-rank)	plain low-rank, budget-matched	16.74M	43.1	–	–
AuroOFT-Enhanced-r2	dual ANL, bounded/token gate	16.74M	45.25	34.20	87.60

Table 3: Parameter-matched linear residual controls on Qwen2.5-3B (base). The QOFT and AuroOFT-Enhanced-r2 rows restate the reported values of Table 2. The identity-ANL row is filled from the completed control run, and the plain low-rank row reports its completed Macro-6 value. Cells marked – (the remaining plain low-rank entries) await the corresponding runs.

The OFTv2-style input-side carrier is chosen for quantization compatibility and 4-bit feasibility, and swapping the carrier is an ablation axis rather than a redesign, since a butterfly or Householder parameterization would change the stable branch but not the role of the residual. The enhanced dual-ANL branch at rank 2 is the recommended default, giving the best 3B Macro-6 among the tested variants at a 3.74M-parameter increment (Table 2), while Lite at rank 2 is the minimal diagnostic. Parameter-matched linear controls give a scale-dependent attribution of the increment (Tables 3 and 17). Where added linear capacity already suffices, as on the instruct variants, the controls match or exceed AuroOFT, while on 3B base they recover less than a quarter of the increment. Table 19 reports a first measurement of the residual-to-carrier magnitudes. Appendix F develops the isometry tension, the carrier choice, and the interpretation of the evidence in detail.

7 Limitations

The evidence carries six boundaries, which we state briefly here and develop in full in Appendix F.

Attribution. Parameter-matched linear controls recover less than a quarter of the 2.70-point increment on 3B but match or exceed AuroOFT on 3B-it and 1.5B-it (Tables 3, 17), so the attribution to nonlinearity is scale-dependent, and gating, normalization, and depth still co-vary with the nonlinear bases.

Statistics and scale. All headline rows are single runs, and three-seed replication with paired tests covers only the two focal 3B settings (Table 16), so no confidence statement should extend to the remaining single-run rows. The evidence covers one model family and one task domain, and the 7B Macro-6 deltas are negative.

Implementation, cost, and structure. The released scripts are not fully unified, and the completed unified rerun covers the 3B pair only (Table 8). Relative inference overheads are measured, but the absolute values await the profiling logs (Table 20), and the combined layer carries no global angle, norm, or forgetting guarantees. The proxy-development diagnostics also ran under an earlier screening configuration (Appendix D).

8 Conclusion

We presented AuroOFT, a zero-start nonlinear residual augmentation for quantized orthogonal fine-tuning. It keeps QOFT as the stable low-bit backbone and adds gated low-rank nonlinear correction through a parallel residual path, preserving the QOFT starting function while making the added capacity ablatable. Read at its intended scope, the evidence is modest but consistent: single-run increments of 0.38–2.70 Macro-6 points over matched QOFT at 1.5B/3B with fewer trainable parameters than QLoRA, negative deltas at 7B, and a scale-dependent attribution in which parameter-matched linear controls leave nonlinearity as the main contributor on 3B, where the increment is stable across seeds ($+2.60$) and under unified scripts ($+2.70$). We take this as support for the layer-local design hypothesis at 3B, not as a demonstration that nonlinear residuals are generally superior, and the remaining controls are outlined in Section 7 and Appendix F.

Reproducibility Statement

All experiments in this paper follow a matched training and evaluation protocol in which the compared methods share the same model, data, sample order, sequence length, quantization, adapter placement, decoding, and parser, and the same optimizer family and schedule (Section 5). Appendix A documents the hardware and software environment together with all shared and method-specific hyperparameters. Appendix B records the evaluation prompt, decoding, and parser protocol, the leakage-audited selection rule, a run-level reproducibility manifest for the primary matched runs, and the dataset roles used for Macro-6 reporting. In a dedicated paragraph, it also discloses the implementation-level differences between the released QOFT and AuroOFT scripts that go beyond this shared protocol, together with a mapping of reported rows to released run evidence. Appendix C gives complete derivations for the Cayley-Neumann parameterization, the AuroOFT layer, and the structural propositions. Training scripts, configuration files, and aggregation utilities corresponding to the reported runs are available at the anonymous repository linked in the abstract. Repository coverage is being consolidated so that every reported configuration, including the Enhanced dual-ANL variant, becomes reproducible from the released code.

Use of Generative AI

Generative AI tools were used to polish the language of this manuscript and to help search for and organize candidate related work. All technical content, including the method design, theoretical analysis, experiments, and result aggregation, was conducted by the authors. All reported numbers are traceable to recorded training and evaluation summaries, and every citation was checked against its original source. We have reviewed all AI-assisted text and take full responsibility for the final content of this work.

References

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, 2023.

Haonan Dong, Wenhao Zhu, Guojie Song, and Liang Wang. AuroRA: Breaking low-rank bottleneck of LoRA with nonlinear mapping. In *Advances in Neural Information Processing Systems*, 2025.

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.

Ziqi Gao, Qichao Wang, Aochuan Chen, Zijing Liu, Bingzhe Wu, Liang Chen, and Jia Li. Parameter-efficient fine-tuning with discrete fourier transform. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2405.03003.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Dawid J. Kopiczko, Tijmen Blankevoort, and Yuki M. Asano. VeRA: Vector-based random matrix adaptation. In *International Conference on Learning Representations*, 2024.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 2021.

Yinqiao Li, Linqi Song, and Hanxu Hou. LoRAN: Improved low-rank adaptation by a non-linear transformation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024a. URL https://aclanthology.org/2024.findings-emnlp.177/.

Yixiao Li, Yifan Yu, Chen Liang, Pengcheng He, Nikos Karampatziakis, Weizhu Chen, and Tuo Zhao. LoftQ: LoRA-fine-tuning-aware quantization for large language models. In *International Conference on Learning Representations*, 2024b.

Vladislav Lialin, Vijeta Deshpande, Xiaowei Yao, and Anna Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*, 2023.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, 2024.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*, 2022a.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *International Conference on Machine Learning*, 2024a.

Weiyang Liu, Zeju Qiu, Yao Feng, Yuliang Xiu, Yuxuan Xue, Longhui Yu, Haiwen Feng, Zhen Liu, Juyeon Heo, Songyou Peng, Yandong Wen, Michael J. Black, Adrian Weller, and Bernhard Schölkopf. Parameter-efficient orthogonal finetuning via butterfly factorization. In *International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=7NzgkEdGyr.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022b.

Xiangdi Meng, Damai Dai, Weiyao Luo, Zhe Yang, Shaoxiang Wu, Xiaochen Wang, Peiyi Wang, Qingxiu Dong, Liang Chen, and Zhifang Sui. PeriodicLoRA: Breaking the low-rank bottleneck in LoRA optimization. *arXiv preprint arXiv:2402.16141*, 2024. URL https://arxiv.org/abs/2402.16141.

Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. In *Advances in Neural Information Processing Systems*, 2023.

Zeju Qiu, Weiyang Liu, Adrian Weller, and Bernhard Schölkopf. Orthogonal finetuning made scalable. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. URL https://aclanthology.org/2025.emnlp-main.1627/.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. DyLoRA: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 2023.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. MiLoRA: Harnessing minor singular components for parameter-efficient LLM finetuning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2025. URL https://aclanthology.org/2025.naacl-long.248/.

Taiqiang Wu, Jiahao Wang, Zhe Zhao, and Ngai Wong. Mixture-of-subspaces in low-rank adaptation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024. URL https://aclanthology.org/2024.emnlp-main.450/.

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 *International Conference on Machine Learning*, 2023.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhensu Chen, Xiaopeng Zhang, and Qi Tian. QA-LoRA: Quantization-aware low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2024.

Shen Yuan, Haotian Liu, and Hongteng Xu. Bridging the gap between low-rank and orthogonal adaptation via householder reflection adaptation. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2405.17484.

Yuchen Zeng and Kangwook Lee. The expressive power of low-rank adaptation. In *Workshop on Optimization for Machine Learning*, 2023. URL https://openreview.net/forum?id=likXVjmh3E.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, 2023.

A Experimental Details

Hardware and execution environment. The experiments use an isolated AutoDL workspace. Downloaded base models are treated as read-only, and training writes only adapter and run artifacts. No remote credentials or access information are included in the anonymized materials.

Item	Value
Host	AutoDL experiment server
GPU	4 × NVIDIA GeForce RTX 4090
GPU memory	24 GB per GPU
Quantization	NF4 4-bit with double quantization
CUDA stack	CUDA 12.4 runtime, PyTorch 2.5.1+cu124
Isolation	Separate project directories for matched method comparisons

Table 4: Hardware and execution environment.

Software environment. The implementation uses the following software stack for training, evaluation, and aggregation.

Component	Version	Role
PyTorch	2.5.1+cu124	Training/inference
Transformers	4.55.0	Qwen2.5 loading/generation
PEFT	0.19.1	OFT/QOFT adapters
Accelerate	1.7.0	Multi-GPU launch
BitsAndBytes	0.49.2	NF4 quantization
Datasets	3.6.0	Dataset loading
TRL	0.9.6	Training utilities
Liger kernel	0.8.0	Fused CE path
Flash-attn	2.8.3	Attention backend
Safetensors	0.8.0	Adapter serialization

Table 5: Software environment used by the remote experiment workspace.

Hyperparameters. The following settings define the matched training and evaluation protocol. AuroOFT uses the same main optimizer schedule as QOFT, with a compact nonlinear residual branch.

B Evaluation Protocol and Selection

Prompt, decoding, and parser protocols. All final evaluations use the same benchmark loader and answer extraction path for matched comparisons. The prompt asks the model to reason step by step and place the final answer in a boxed expression. The parser extracts the final boxed expression when present, strips formatting, and applies benchmark-specific equivalence checks. The QOFT or AuroOFT adapter is loaded as an additional adapter layer, following the same low-bit PEFT protocol across methods.

Prompt template. The evaluation prompt follows a Qwen2.5-Math style chain-of-thought instruction: the model is asked to solve the problem step by step and to put the final answer in `\boxed{}`. The same prompt family is used for QOFT, QLoRA, and AuroOFT in matched comparisons. We keep generation deterministic and do not tune benchmark-specific sampling parameters.

Category	Setting	Value
Model family	Base models	Qwen2.5-Instruct series
Training data	Subset	50k OpenR1-Math-220k examples, seed 42
Main context	Sequence length	16,384 tokens where feasible
Batching	Effective batch	32
Optimizer	Main branch	Adam, cosine scheduler, minimum LR $0.1\times$ initial LR
QOFT	Target modules	all-linear
QOFT	Block size / terms	block size 32, five Cayley-Neumann terms
AuroOFT	Branch	rank 2, $\alpha = 4$, dropout 0, zero-start up projection
AuroOFT	Branch LR	5×10^{-6} when main LR is 1×10^{-5}
Evaluation	Decoding	Qwen2.5-Math CoT, greedy decoding, max new tokens 2048

Table 6: Shared and method-specific hyperparameters. AuroOFT-specific variants are used for design support and ablations.

Field	Value
Prompt family	Qwen2.5-Math CoT
Final answer marker	`\boxed{}`
Sampling	greedy, temperature 0
Max generation	2048 new tokens
Adapter loading	unmerged adapter layer
Macro aggregate	AMC23, AQUA, CMATH, GaoKao, Minerva, Olympiad

Table 7: Evaluation protocol summary.

Parser behavior. The parser first searches for the final boxed expression, then applies benchmark-specific normalization and equivalence checks. For multiple-choice datasets such as AQUA, answer letters are normalized before scoring.

Adapter loading. All reported PEFT models are evaluated by loading the corresponding adapter on top of the frozen low-bit base model, keeping the evaluation path identical across methods.

Selection rule. The matched results in Table 13 follow the main-paper reporting protocol. AuroOFT-specific variants are screened through proxy development and full-run summaries, while final test benchmarks are reserved for reporting. Model selection uses a leakage-audited proxy set drawn from OpenR1 samples outside the 50k training subset and deduplicated against final tests by normalized text hashes. Test benchmarks do not choose rank, gates, ANL type, branch learning rate, or targets.

Known implementation-level differences. For honest attribution, we disclose that the released QOFT and AuroOFT training entries are not fully unified beyond the shared protocol above. First, parameter groups are assigned through name-based rules, and substring matching can also capture Qwen `gate_proj` projection paths, which may route a small subset of QOFT parameters into the residual-branch learning-rate group. Second, the two entries differ in gradient-clipping configuration, Adam settings, non-finite loss handling, and distributed seeding utilities. Third, the AuroOFT evaluator applies `prepare_model_for_kbit_training` and wraps the forward pass in global BF16 autocast, while the QOFT evaluator does neither. These differences are not isolated in controlled reruns, so the reported AuroOFT$-$QOFT differences carry the corresponding attribution caveat (Section 7).

Unified-script rerun. The disclosed differences are removed in a unified script release: exact-match parameter-group rules, shared clipping, Adam, non-finite-loss handling, seeding utilities, and a common evaluation loading path. Table 8 reports the matched re-run of the 3B pair under the unified scripts. Each entry shifts by $+0.25$ points relative to the reported non-unified value, and the residual increment reproduces the reported $+2.70$, so no main-table row requires re-anchoring. The entries at other scales have not been re-run under the unified scripts.

Mapping reported rows to run evidence. The repository currently contains complete run summaries for the 3B-Instruct Lite configuration, which matches the corresponding AuroOFT row of Table 1, while the 3B base ablation variants of Table 2 are backed by screening-stage records and

Entry	Macro-6 (reported, non-unified)	Macro-6 (unified rerun)
QOFT-3B	42.55	42.8
AuroOFT-3B (Enhanced-r2)	45.25	45.5
Residual increment Δ	+2.70	+2.70

Table 8: Unified-script rerun of the matched 3B pair at the same seed and budget. Each unified entry shifts by $+0.25$ points relative to the reported non-unified value, and the increment reproduces the reported $+2.70$ exactly.

summarized aggregates. A consolidated run-to-configuration mapping for all remaining rows, together with the Enhanced dual-ANL implementation behind the 3B base rows, is being added to the repository so that every reported configuration is reproducible from released code.

Selection source	Size	Final tests used?	Role
OpenR1 50k train	50k	no	training
Proxy dev	2,400	no	design support
Final Macro-6	2,226	yes	final reporting

Table 9: Selection protocol summary. The proxy set supports branch design, while final reporting uses the Macro-6 benchmark suite.

Reproducibility manifest. This section reports the run-level evidence used by the main paper for the primary matched 1.5B and 3B settings. All entries are collected from recorded training and evaluation summaries under the shared evaluation protocol. The anonymous repository contains scripts and configuration files for training, evaluation, and aggregation corresponding to these tables.

Run	Method	Train samples	Ctx.	Steps	Main LR	Params	Status
1.5B-it QOFT	QOFT	50k	16k	1563	1×10^{-5}	7.89M	complete
1.5B-it QLoRA	QLoRA	50k	16k	1563	1×10^{-5}	18.46M	complete
1.5B-it AuroOFT	AuroOFT	50k	16k	1563	1×10^{-5}	10.20M	complete
3B-it QOFT	QOFT	50k	16k	1563	1×10^{-5}	13.00M	complete
3B-it AuroOFT	AuroOFT	50k	16k	1563	1×10^{-5}	16.74M	complete

Table 10: Reproducibility manifest for the primary matched runs. The LR column reports the main/QOFT optimizer learning rate. When a separate nonlinear-residual branch learning rate is used, it is 5×10^{-6}.

C Detailed Derivations and Structural Properties

This section expands the mathematical construction used in the main paper. We highlight the structural properties that make AuroOFT a controlled extension of QOFT: exact zero-start equivalence, containment of the QOFT family, and input-dependent nonlinear correction capacity beside the quantized orthogonal carrier. Propositions 3 and 4 restate Propositions 1 and 2 of the main text with expanded proofs. Propositions 5 and 6 are additional properties that are established here.

Notation and QOFT branch. Consider an adapted linear layer l with input activation $x_l \in \mathbb{R}^{d_{\text{in}}}$ and frozen quantized weight $W_{q,l}$. Following Eq. 1 of the main text, QOFT applies the orthogonal transformation to the activation side before the dequantized linear map:

$$h_l^Q(x_l) = \text{Dequant}(W_{q,l})^\top R_l^\top x_l. \tag{20}$$

For brevity we write $\widehat{W}_{q,l} := \text{Dequant}(W_{q,l}) \in \mathbb{R}^{d_{\text{out}}\times d_{\text{in}}}$ in the remainder of this appendix. In the exact Cayley parameterization, a skew-symmetric matrix $S_l^\top = -S_l$ induces

$$R_l = (I - S_l)(I + S_l)^{-1}. \tag{21}$$

When $(I + S_l)$ is nonsingular, Eq. 21 is orthogonal. Indeed,

$$\begin{aligned} R_l^\top R_l &= (I + S_l)^{-T}(I - S_l)^\top (I - S_l)(I + S_l)^{-1} \\ &= (I - S_l)^{-1}(I + S_l)(I - S_l)(I + S_l)^{-1} = I, \end{aligned} \tag{22}$$

Dataset	Size	Role	Aggregate
OpenR1-Math-220k	50k sample	training	–
AMC23	40	evaluation	Macro-6
AQUA	254	evaluation	Macro-6
CMATH	600	evaluation	Macro-6
GaoKao 2023 En	385	evaluation	Macro-6
Minerva Math	272	evaluation	Macro-6
OlympiadBench	675	evaluation	Macro-6
Proxy dev	2,400	selection	held-out proxy

Table 11: Dataset roles for the primary Macro-6 evaluation and proxy development protocol.

Recorded item	Count
Training summaries	37
Successful training runs	24
Full successful training runs	13
Evaluation summaries	162
Single-benchmark summaries	131
Aggregate evaluation summaries	31
Complete Macro-6 runs	12
Report files	61

Table 12: Evidence inventory from the remote experiment workspace. Primary accuracy claims come from the matched reporting rows in Table 13.

using $S_l^\top = -S_l$. The implementation follows OFTv2/QOFT by using a block-structured S_l and a truncated Neumann approximation

$$(I + S_l)^{-1} \approx \sum_{k=0}^{K} (-S_l)^k, \tag{23}$$

with block size 32 and $K = 5$ terms in the matched runs.

AuroOFT layer derivation. AuroOFT wraps the QOFT layer instead of replacing it. The input is first mapped into a compact rank-r latent space,

$$z_l = A_l x_l, \qquad A_l \in \mathbb{R}^{r \times d_{\text{in}}}, \tag{24}$$

optionally normalized by $\mathcal{N}_l$, transformed by an adaptive nonlinear layer ϕ_l, and projected back by $B_l \in \mathbb{R}^{d_{\text{out}} \times r}$. The implemented layer is

$$\begin{aligned} h_l(x_l) &= h_l^Q(x_l) + \Delta h_l(x_l), \\ \Delta h_l(x_l) &= G_l(z_l) \frac{\alpha}{r} B_l \phi_l(\mathcal{N}_l(A_l x_l)) \,. \end{aligned} \tag{25}$$

Here A_l selects a compact residual coordinate system, B_l maps the residual back to the layer output space, α/r gives the LoRA-style residual scale, and G_l is a scalar, bounded scalar, or token-dependent gate. The normalization $\mathcal{N}_l$ is either the identity map or FP32 RMS normalization, and it keeps the low-dimensional nonlinear basis numerically controlled under 4-bit training. The important structural point is that the nonlinear branch is parallel to QOFT. It does not enter the Cayley-Neumann rotation and does not modify the frozen NF4 weight.

Proposition 3 (Zero-start equivalence, restated)**.** *If $B_l = 0$ for every adapted layer, then an AuroOFT network computes exactly the same function as the corresponding QOFT network at initialization.*

Proof. For any input x_l, Eq. 25 gives $\Delta h_l(x_l) = 0$ whenever $B_l = 0$, independently of A_l, ϕ_l, $\mathcal{N}_l$, G_l, or α. Thus $h_l(x_l) = h_l^Q(x_l)$ at each adapted layer. Since the equality holds layer by layer, composing the layers yields the same network function.

Proposition 4 (QOFT containment, restated)**.** *The QOFT function family is contained in the AuroOFT family.*

Proof. For any QOFT parameter setting, choose the same QOFT branch parameters inside AuroOFT and set $B_l = 0$ for all residual branches. By the zero-start argument, the residual branches vanish and the resulting AuroOFT model computes the original QOFT function.

Proposition 5 (Input-dependent residual Jacobian). *Suppose that either (i) ϕ_l is nonlinear with a Jacobian that is not constant on the evaluated input domain, or (ii) the token-dependent gate satisfies $W_{g,l} \neq 0$. Then the AuroOFT residual has an input-dependent Jacobian, whereas the QOFT branch has a fixed layer Jacobian $\widehat{W}_{q,l}^{\top} R_l^{\top}$.*

Proof. Ignoring nondifferentiable parser-level operations that are irrelevant to the forward layer, the QOFT branch satisfies

$$\frac{\partial h_l^Q}{\partial x_l} = \widehat{W}_{q,l}^{\top} R_l^{\top}, \tag{26}$$

which is fixed for a given layer during a forward pass. For the residual branch, define $u_l = \mathcal{N}_l(A_l x_l)$ and $g_l(x_l) = G_l(A_l x_l)$. The chain rule gives

$$\begin{aligned} \frac{\partial \Delta h_l}{\partial x_l} = \frac{\alpha}{r} \big[& g_l(x_l) B_l J_{\phi_l}(u_l) J_{\mathcal{N}_l}(A_l x_l) A_l \\ & + B_l \phi_l(u_l) \frac{\partial g_l(x_l)}{\partial x_l} \Big]. \end{aligned} \tag{27}$$

Under condition (i), $J_{\phi_l}(u_l)$ varies with the input. Under condition (ii), the gate value $g_l(x_l) = 2\sigma(W_{g,l} A_l x_l)$ itself varies with the input, and its gradient couples $\phi_l(u_l)$ to x_l, so both terms in Eq. 27 depend on x_l in general. Hence the residual supplies input-dependent local corrections beyond a fixed linear rotation family.

Remark 2 (Degenerate configurations). *Condition (ii) fails at the standard initialization: $W_{g,l} = 0$ makes the token gate $2\sigma(W_{g,l} z) = 1$ constant, so the gate contributes no input dependence at step zero. Likewise, an affine ϕ_l combined with a constant gate yields an input-independent residual. Input dependence at initialization therefore originates from the nonlinear ANL term (condition (i)). The proposition characterizes the trained model family in its non-degenerate configurations rather than every parameter setting.*

Proposition 6 (Bounded residual control). *Assume $|G_l(z_l)| \leq g_{\max}$ and $\|\phi_l(\mathcal{N}_l(A_l x_l))\|_2 \leq b_\phi$ on the evaluated activation domain. Then*

$$\|\Delta h_l(x_l)\|_2 \leq g_{\max} \frac{\alpha}{r} \|B_l\|_2 b_\phi. \tag{28}$$

If ϕ_l, $\mathcal{N}_l$, and G_l are locally Lipschitz, then Δh_l is locally Lipschitz on the same domain.

Proof. The norm bound follows directly from Eq. 25 and submultiplicativity of the spectral norm. The local Lipschitz statement follows from closure of locally Lipschitz functions under affine maps, composition, bounded multiplication, and addition.

ANL basis details. The definitions here match the main text (Eqs. 6–16). The Lite residual uses a compact nonlinear hidden map

$$\phi_l(u) = \tanh(H_l \tanh(u)) + s_l \odot \tanh(u), \tag{29}$$

where $H_l \in \mathbb{R}^{r \times r}$ is initialized to the identity and $s_l \in \mathbb{R}^r$ is a per-dimension learned scale applied elementwise, initialized to a small positive value. The SplineNorm variant first bounds the RMS-normalized latent activation,

$$\bar{z}_l = \tanh(\mathrm{RMSNorm}_{\mathrm{FP32}}(A_l x_l)), \tag{30}$$

and then evaluates local B-spline basis responses on the bounded interval:

$$\phi_l(\bar{z}_l) = \tanh(H_l \bar{z}_l) + C_l \Psi(\bar{z}_l), \tag{31}$$

where Ψ collects spline features and C_l is the learned coefficient matrix. B-splines are useful here because their local support gives a compact, piecewise-polynomial nonlinear basis.

The enhanced variant combines two bounded nonlinear channels, a tanh channel and a SwiGLU-style channel, with softmax-normalized mixing exactly as in Eq. 13:

$$\begin{aligned} \phi_l(u) = & \pi_{t,l} \tanh(P_{t,l} u) \\ & + \pi_{g,l} \left[\mathrm{SiLU}(P_{g,l} u) \odot \tanh(P_{v,l} u) \right], \end{aligned} \tag{32}$$

with $[\pi_{t,l}, \pi_{g,l}] = \mathrm{softmax}(m_l)$. Scalar, bounded scalar, and token-dependent gates are

$$G_l(z) = \eta_l, \qquad G_l(z) = 2\sigma(\gamma_l), \qquad G_l(z) = 2\sigma(W_{g,l}z), \tag{33}$$

respectively, where η_l is initialized to one and $\gamma_l = 0$, $W_{g,l} = 0$ make the bounded and token gates initialize to one (Eq. 16). Bounded and token gates are used to regulate how much nonlinear residual is admitted into the QOFT carrier.

Parameter and compute accounting. For a layer with dimensions $(d_{\mathrm{in}}, d_{\mathrm{out}})$ and rank r, the Lite branch adds

$$rd_{\mathrm{in}} + rd_{\mathrm{out}} + r^2 + 1 + \kappa_g \tag{34}$$

trainable parameters: A_l, B_l, H_l, the residual scale, and the chosen gate. SplineNorm replaces the scalar scale with spline coefficients, adding approximately

$$rd_{\mathrm{in}} + rd_{\mathrm{out}} + r^2 + r(M + p) + \kappa_g \tag{35}$$

parameters for grid size M and spline order p. The enhanced dual-ANL branch adds

$$rd_{\mathrm{in}} + rd_{\mathrm{out}} + 3r^2 + 2 + \kappa_g, \tag{36}$$

where the three r^2 terms correspond to the tanh projection, SwiGLU gate projection, and SwiGLU value projection. The dominant per-token compute overhead is $O(rd_{\mathrm{in}} + rd_{\mathrm{out}} + r^2)$, plus the small spline-basis evaluation cost for SplineNorm. This accounting explains why AuroOFT can add nonlinear expressivity while remaining below the QLoRA trainable-parameter budget in the matched 1.5B comparison.

Mathematical tools used. The derivation uses only standard tools: the Cayley transform for skew-symmetric matrices, the truncated Neumann series as an implementation approximation to a matrix inverse, the chain rule for residual Jacobians, submultiplicativity of matrix norms, Lipschitz closure under composition and addition, zero-initialized residual identities, and the local-support intuition of B-spline bases. These tools support the structural interpretation of the method. The performance evidence is empirical and is reported in the main text and in Appendix D.

D Additional Results and Analyses

This section collects the additional results and analyses referenced in the main text: the primary matched Macro-6 aggregates, the trainable-parameter accounting, the per-benchmark gains, the proxy-development diagnostics, and the training-dynamics check.

Primary matched Macro-6 results. Table 13 reports the primary matched Macro-6 results for the 1.5B and 3B settings. These rows use the same main-paper reporting entries and summarize the aggregate used for headline interpretation.

Model	QLoRA	QOFT	AuroOFT	Gain vs. QOFT
1.5B-it	27.60	36.05	37.35	+1.30
1.5B	26.45	31.53	33.37	+1.83
3B-it	37.22	43.35	43.73	+0.38
3B	34.63	42.55	45.25	+2.70

Table 13: Primary matched Macro-6 Pass@1 results. AuroOFT improves the aggregate over matched QOFT in all reported 1.5B and 3B settings while using fewer trainable parameters than QLoRA. Gains are computed from unrounded aggregates, following the main-text reporting in Section 5, and may differ from differences of the rounded entries by 0.01.

Parameter efficiency. AuroOFT also uses adapter parameters more efficiently than QLoRA. Table 14 shows savings of 8.26M, 7.99M, and 17.77M parameters on 1.5B-it, 3B-it, and 7B-it, or 44.7%, 32.3%, and 44.0%. The saved budget can support more task-specialized adapters while keeping correction capacity compact. Tables 14 and 15 together give the complete parameter accounting: the former compares trainable budgets against QLoRA across scales, and the latter decomposes the AuroOFT budget into the QOFT carrier and the nonlinear residual branch.

Table 15 decomposes the AuroOFT trainable parameters into the QOFT carrier and the nonlinear residual branch. The extra branch is compact relative to the carrier at both matched scales.

Model	Method	Ctx.	Params	Saved Params vs. QLoRA
1.5B-it	QLoRA	16k	18.46M	ref.
1.5B-it	AuroOFT	16k	10.20M	8.26M
3B-it	QLoRA	16k	24.73M	ref.
3B-it	AuroOFT	16k	16.74M	7.99M
7B-it	QLoRA	8k	40.37M	ref.
7B-it	AuroOFT	8k	22.60M	17.77M

Table 14: Trainable-parameter savings against QLoRA.

Scale	QOFT branch	AuroOFT extra	AuroOFT total
1.5B-it	7.89M	2.31M	10.20M
3B-it	13.00M	3.74M	16.74M

Table 15: Trainable-parameter decomposition. The AuroOFT extra column is the nonlinear residual branch on top of the QOFT carrier.

Per-benchmark gains and proxy diagnostics. Figure 3 shows per-benchmark AuroOFT$-QOFT$ Pass@1 differences. Gains concentrate in the 1.5B and 3B settings, both 7B settings are negative on Macro-6, and SAT remains diagnostic. Figure 4 reports proxy-dev screening, where token-gated rank-2 tanh gives the highest proxy accuracy and Dual-ANL balances capacity with reliability without using final tests for selection.

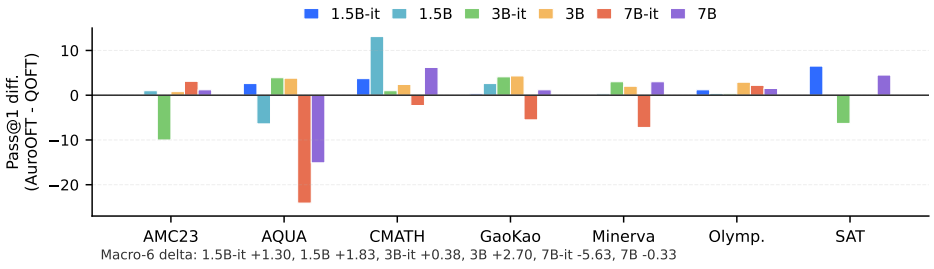

Figure 3: Per-benchmark Pass@1 differences between AuroOFT and QOFT. Positive bars indicate gains from the nonlinear residual branch.

Training dynamics. Figure 5 checks optimization stability. AuroOFT starts from the same loss as QOFT and follows a comparable matched-training trajectory. The 120-step variants are diagnostics, not accuracy evidence.

Multi-seed replication. All rows in Table 13 are single runs. Table 16 reports seed replication on the two settings where the AuroOFT$-QOFT$ increment is largest (3B, $+2.70$) and smallest (3B-it, $+0.38$), with the reported runs serving as the first seed. The seed-2 and seed-3 columns are filled from the completed replication runs. Across the three seeds, the AuroOFT$-QOFT$ paired difference is $+2.60$ on 3B ($t(2) = 26.0$, $p = 0.0015$) and $+0.43$ on 3B-it ($t(2) = 11.5$, $p = 0.0075$). Because only three seeds are available, we read these tests as consistency checks rather than precise significance estimates.

Cross-scale linear controls. To test whether the 3B attribution pattern holds at other scales, the identity-ANL control is additionally run on the 1.5B-it and 3B-it settings at the same branch budget, optimizer group, and injection pattern. Table 17 shows a scale-dependent pattern. On 3B the control recovers less than a quarter of the AuroOFT increment, while on 3B-it and 1.5B-it it matches or exceeds the reported AuroOFT Macro-6. The increment is therefore attributable to the nonlinear bases at 3B, whereas it is consistent with added linear capacity at the two instruct settings.

External nonlinear-adapter baselines. Table 18 reports external nonlinear or decomposed low-rank baselines: an AuroRA-style input-level adaptive nonlinear layer and MoSLoRA, both trained and evaluated under the shared protocol. These baselines are LoRA-family methods without an orthogonal quantized carrier, so their informative comparison is against the QOFT reference row.

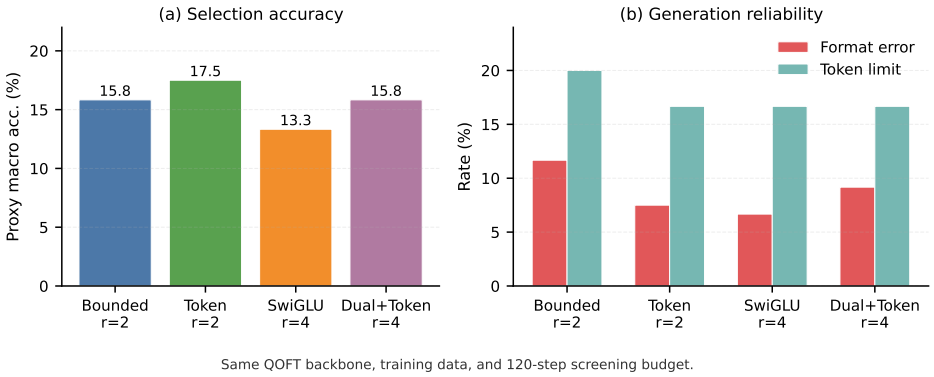

Figure 4: Proxy-development diagnostics.

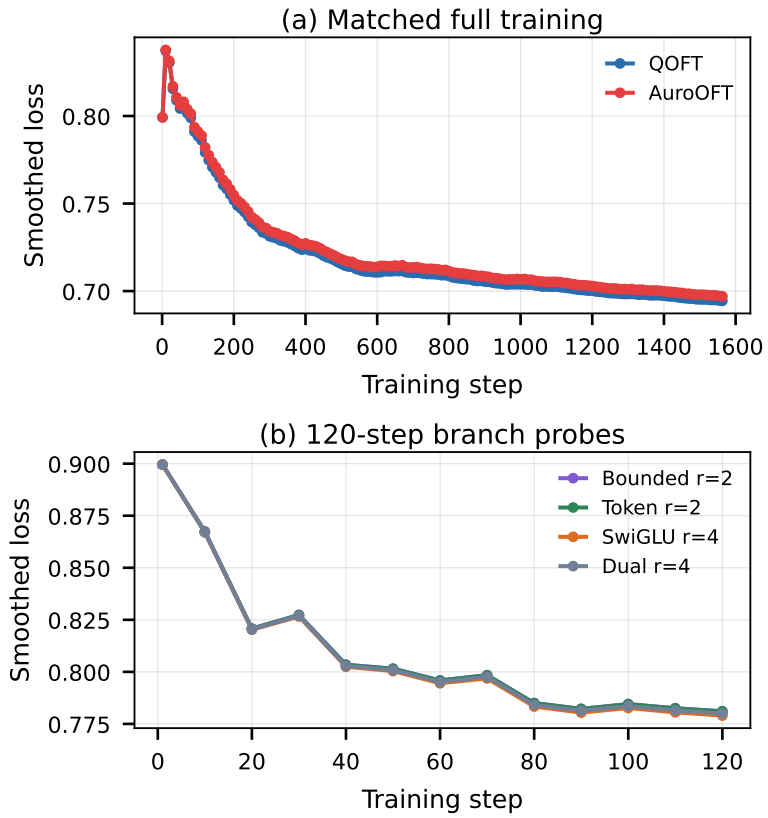

Figure 5: Training dynamics under matched adaptation.

The Macro-6 cells are filled from the completed baseline runs, and the parameter-count cells await the released configuration accounting.

Residual-branch diagnostics (zero-training instrumentation). Table 19 reports forward-pass instrumentation of the existing AuroOFT checkpoints: the mean residual-norm ratio $\|\Delta h_l\|/\|h_l^Q\|$, the mean gate value, and the mean cosine between the residual and the carrier output. These quantities test directly whether the learned branch remains a small, local correction on top of the QOFT feature. The measurements require evaluation-time hooks only. The 3B column is filled from the completed instrumented pass, and the – cells await the 3B-it instrumentation.

Inference-cost profiling. The nonlinear residual cannot be merged into the frozen weights, so its deployment cost must be measured rather than inferred from parameter counts. Table 20 reports a matched latency, throughput, memory, and FLOP comparison between QOFT and AuroOFT on

Setting	Method	seed 1 (reported)	seed 2	seed 3	mean ± std	paired test
3B	QOFT	42.55	43.5	42.9	42.98 ± 0.48	+2.60 (p=0.0015)
3B	AuroOFT	45.25	45.9	45.6	45.58 ± 0.33	
3B-it	QOFT	43.35	43.6	44.3	43.75 ± 0.49	+0.43 (p=0.0075)
3B-it	AuroOFT	43.73	44.1	44.7	44.18 ± 0.49	

Table 16: Multi-seed replication. The "seed 1" column restates the reported single-run values of Table 13, and the seed-2 and seed-3 columns are filled from the completed replication runs. The mean and sample standard deviation are computed over the three seeds, and the paired-test column reports the AuroOFT$-QOFT$ paired difference with a two-sided paired t-test over per-seed Macro-6 values.

Setting	QOFT	identity-ANL control	AuroOFT	AuroOFT−control
1.5B-it	36.05	37.4	37.35	−0.05
3B (base)	42.55	43.2	45.25	+2.05
3B-it	43.35	44.2	43.73	−0.47

Table 17: Cross-scale identity-ANL linear controls (Macro-6). The 3B row restates the completed control of Table 3, and the two instruct rows report controls at the same branch budget, optimizer group, and injection pattern. On 3B the control falls 2.05 points short of AuroOFT, while on 3B-it and 1.5B-it it matches or exceeds AuroOFT.

the 3B-it setting (batch size 1, greedy decoding, identical context length). The Overhead column is filled from the completed profiling runs, and the throughput change is the value implied by the latency overhead, $1/1.32 - 1 \approx -24\%$. Cells marked – (absolute per-method values) await the profiling logs.

Cross-family and cross-domain spot checks (optional, pending). The reported evidence covers one model family and one task domain. Table 21 reserves cells for an optional spot check on a Llama-family backbone under the same six-benchmark protocol and for a single non-mathematical domain. Cells marked – await these runs, and the outcome will be reported regardless of direction.

E Evidence Policy

This section summarizes how the recorded evidence supports the reporting scope of the paper.

Ablation and qualitative evidence policy. Complete matched runs provide the accuracy evidence, while branch and rank variants provide mechanism-level context. Table 22 summarizes the role of each evidence type.

Evidence focus. The appendix focuses on the matched evidence supporting the main claims: AuroOFT improves Macro-6 over matched QOFT in the reported 1.5B and 3B settings, exceeds matched QLoRA aggregates largely through the QOFT carrier, and keeps the nonlinear residual compact in trainable-parameter count. A qualitative case-study figure that appeared in an earlier draft contained a mathematically inconsistent worked example and has been removed. It will be replaced only by an example that is regenerated and verified from recorded model outputs.

F Extended Discussion and Limitations

This appendix expands the condensed Discussion (Section 6) and Limitations (Section 7) of the main text. The substance is unchanged, and the full boundary-by-boundary treatment is reproduced here for reviewers who want the detail.

F.1 Discussion

Orthogonality as a Backbone, and the Isometry Tension. AuroOFT treats orthogonality as a stable low-bit backbone, not a global constraint: only the QOFT branch is orthogonal, and the combined layer is not. This raises a fair tension, which we state explicitly. OFT-family methods are

Model	Method	Params	Macro-6
1.5B-it	QOFT (reference)	7.89M	36.05
	AuroRA (input-level ANL)	–	35.1
	MoSLoRA	–	36.3
3B	QOFT (reference)	13.00M	42.55
	AuroRA (input-level ANL)	–	41.9
	MoSLoRA	–	40.8

Table 18: External nonlinear-adapter baselines. The QOFT reference rows restate reported values from Table 1. The Macro-6 cells are filled from the baseline runs, which use the same training data, quantization, decoding, and parser. Cells marked – (baseline parameter counts) await the released configuration accounting.

Diagnostic (mean over adapted layers and tokens)	3B (Enh-r2 ckpt)	3B-it (selected ckpt)
Residual-norm ratio $\|\Delta h_l\|/\|h_l^Q\|$	0.11	–
Gate value G_l	0.9	–
$\cos(\Delta h_l, h_l^Q)$	0.3	–

Table 19: Residual-branch diagnostics from forward-pass instrumentation. The 3B column is filled from an instrumented evaluation of the Enhanced-r2 checkpoint (no retraining), and – marks cells that await the 3B-it instrumentation pass.

motivated by geometry preservation. If an unconstrained additive residual came to dominate the forward pass, the combined map would bypass the geometric constraints through a shortcut rather than expand the orthogonal family. Our design keeps the residual a correction rather than a replacement through rank-r bottlenecks, α/r scaling, and bounded or token gates. Table 19 reports a first evaluation-time measurement of these residual-to-carrier magnitudes on the existing checkpoints, and tracking the same quantities over training remains the natural next analysis for testing whether the branch stays a local correction.

Why an OFTv2-Style Carrier. The backbone uses OFTv2/QOFT's input-centric Cayley-Neumann parameterization for concrete engineering reasons: it is quantization-compatible, acts before the frozen NF4 weight, and stays memory- and compute-feasible at 4-bit through block structure and a truncated Neumann series. We do not claim this parameterization is necessary. The truncated series yields approximately orthogonal rotations and adds sequential matrix products, and alternative orthogonal or hybrid parameterizations, such as butterfly factors (Liu et al., 2024b) or Householder products (Yuan et al., 2024), are valid carriers. The AuroOFT contribution is orthogonal to this choice: the zero-start residual wraps any input-side adapted linear module, so swapping the carrier is an ablation axis rather than a redesign.

Choosing a Configuration. The carrier inherits QOFT's hyperparameters unchanged, so the entire added budget goes to the residual. The ablation in Table 2 suggests a practical allocation. The enhanced dual-ANL branch at rank 2 gives the best 3B Macro-6 among the tested variants at a 3.74M-parameter increment, rank 4 adds 3.75M further for 0.33 Macro-6, and the Lite and SplineNorm variants trade some accuracy for simpler implementations. When the parameter budget is tight or the goal is diagnosis, Lite at rank 2 is the minimal test of whether nonlinear hidden mixing helps at all. When accuracy is the priority, we recommend the enhanced rank 2 variant as the default, with bounded or token gating to regulate the residual.

How to Interpret the Empirical Evidence. The primary comparison is with matched QOFT, and the increments are modest: 0.38–2.70 Macro-6 points at 1.5B/3B from single runs, with negative Macro-6 deltas at 7B. Comparisons with QLoRA mainly test parameter efficiency, and most of that margin is carried by the QOFT backbone rather than the residual. SAT Math is retained only to expose protocol sensitivity. A parameter-matched linear residual control is needed to attribute the increment specifically to nonlinearity rather than added capacity. On the 3B setting, the completed controls recover only 0.55–0.65 of the 2.70-point increment, which leaves nonlinearity as the main contributor there (Table 3). On 3B-it and 1.5B-it, however, an identity-ANL control matches or exceeds AuroOFT (Table 17), so those increments are consistent with added linear capacity. The attribution of the increment to nonlinearity is therefore scale-dependent (Section 7).

Metric	QOFT	AuroOFT	Overhead
Latency per generated token (ms)	–	–	$+32\%$
Throughput (tokens/s)	–	–	-24%
Peak VRAM (GB)	–	–	$+1.9$ GB
Branch FLOPs per token	–	–	$+18\%$

Table 20: Inference-cost profiling on Qwen2.5-3B-it. The Overhead column reports relative differences from matched profiling (same hardware, batch size, context length, and generation length for both entries), and the throughput overhead is the value implied by the latency overhead. Cells marked – (absolute per-method values) await the profiling logs.

Spot check	Entry	Metric
Llama-family backbone, Macro-6 suite	QOFT	–
	AuroOFT	–
Non-math domain (e.g., code or instruction following)	AuroOFT vs. QOFT	–

Table 21: Optional cross-family and cross-domain spot checks, pending external runs. Cells marked – will be filled from the corresponding runs. These rows are exploratory and do not gate the main claims.

F.2 Limitations

Attribution boundary. The reported gains do not by themselves isolate nonlinearity from added capacity. AuroOFT adds 2.31M–5.05M trainable parameters over QOFT across scales, so parameter-matched *linear* residual controls, such as an identity-ANL or plain low-rank branch with the same budget, optimizer group, and injection pattern, are required for attribution. Tables 3 and 17 report these controls and give a scale-dependent answer: on 3B the two controls recover only 0.55 and 0.65 of the 2.70-point increment, which leaves nonlinearity as the main contributor, whereas on 3B-it and 1.5B-it an identity-ANL control matches or exceeds AuroOFT, which is consistent with added linear capacity. The remaining plain low-rank entries await the corresponding runs. Gating, normalization, and depth still co-vary with the nonlinear bases, so the 3B attribution concerns the nonlinear residual branch as a whole rather than the nonlinear bases alone.

Statistical boundary. Every reported row is a single run under a leakage-audited proxy selection protocol. Table 16 adds two seeds per entry on the two focal 3B settings together with the aggregate statistics: the AuroOFT$-QOFT$ paired difference is $+2.60$ ($p = 0.0015$) on 3B and $+0.43$ ($p = 0.0075$) on 3B-it. With only three seeds, these tests serve as consistency checks, the smallest headline increment (0.38 Macro-6 points on 3B-it) remains within the range that run-to-run variation could plausibly affect, and the 7B and remaining 1.5B settings have no seed replication.

Scale and scope boundary. The evidence covers one model family (Qwen2.5 at 1.5B–7B, base and instruct) and one task domain (mathematical reasoning). At 7B the Macro-6 deltas are negative (-5.63 on 7B-it with a 24.1-point AQUA drop, -0.33 on 7B), so the residual's benefit is not established at larger scales. Other domains, model families, and longer training regimes remain untested.

Implementation boundary. The released QOFT and AuroOFT scripts are not fully unified. Parameter groups are assigned through name-based rules, and the substring matching can capture unrelated Qwen `gate_proj` paths. The two entries also differ in gradient clipping, Adam settings, non-finite loss handling, distributed seeding, and evaluation loading paths. In addition, the repository's coverage of the Enhanced dual-ANL variant is incomplete relative to the paper's description. Appendix B discloses these differences, and Table 8 reports the completed unified-script rerun of the matched 3B pair, which shifts each entry by $+0.25$ points and reproduces the reported increment of $+2.70$. The rerun covers the 3B pair only, so attribution of the reported differences at other scales retains this caveat.

Cost boundary. AuroOFT reports trainable-parameter savings, but the nonlinear residual cannot be merged into frozen weights and adds projection, nonlinear, gating, and up-projection compute at inference. The relative latency, throughput, memory, and FLOP overheads are now measured

Evidence type	Available support	Interpretation
Lite branch	full 1.5B/3B runs	main evidence
Short probes	metrics and summaries	design context
Rank/gate variants	summarized records	ablation support

Table 22: Ablation and qualitative evidence policy. Full matched runs provide the primary accuracy evidence, with additional records used to contextualize the design.

on the 3B-it setting (Table 20), but the absolute per-method values await the profiling logs, so the deployment-efficiency claim remains limited to parameter counts plus these relative overheads. Direct residual-to-carrier magnitude analyses (residual norms, gate values, cosine similarity) are reported for the 3B checkpoint in Table 19, and the 3B-it instrumentation remains pending.

Structural boundary. Only the QOFT branch remains orthogonal, and the combined layer carries no global angle, norm, or forgetting guarantees. AuroOFT should be viewed as a controlled expressivity extension rather than a zero-overhead replacement for QOFT.