MULTI-ANCHOR QK-COUPLING IN LLMS: SHARED ATTENTION ROUTING AND EFFICIENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We identify multi-anchor QK-coupling, an anchor-relative geometry in which query and key trajectories remain coupled across layers, revealing shared routing structure in trained LLMs. We analyze this observation from three complementary perspectives: shared relational reconstruction error is jointly controlled by the common component's spectral tail and the mismatch measured by multi-anchor coupling; under spectral-dominance and relational-orthogonality conditions, common-core-preserving MLA-style compression can increase coupling; and private residual branches provide complementary first-order score-space correction directions around a fixed shared representation. Guided by these results, we propose a QK-coupled parameterization with a shared routing trunk and lightweight query-private/key-private residual branches. The design supports standard pre-training, MLA-style compressed-KV pre-training, and LoRA-style fine-tuning, reducing QK projection parameters, projection FLOPs, and training-state storage, with additional latent KV-cache savings on the evaluated SmolLM-style MLA backbones. Experiments show improved or competitive quality, including 75% QK-side FLOP/optimizer-memory savings with better perplexity in standard pre-training and a 50.57% QK-path projection-FLOP reduction with better final perplexity in a 3B, 200B-token MLA-style run.

1 INTRODUCTION

Large language models (LLMs) have entered a regime where progress is heavily driven by the joint scaling of model parameters, training data, context length, and computational budget (Kaplan et al., 2020; Hoffmann et al., 2022; Brown et al., 2020; Touvron et al., 2023; Liu et al., 2024b; Yang et al., 2025; DeepSeek-AI, 2026). In this regime, the Transformer block (Vaswani et al., 2017) acts as the fundamental economic unit of the model; any local structural over-parameterization multiplies across sequence length, depth, and the entire training horizon, quickly escalating into a massive system-level cost. Therefore, identifying and reducing structural redundancies within the Transformer complements hardware and kernel-level advances in efficient pre-training and fine-tuning.

Operator-aware allocation of training resources. Current broad efficiency paradigms—such as optimizer-state partitioning (Rajbhandari et al., 2020; Shazeer & Stern, 2018; Dettmers et al., 2021; Zhao et al., 2024; He et al., 2024; Chen et al., 2025; Robert et al., 2024; Zhu et al., 2025), architecture-level reduction (Hao et al., 2024; Liu et al., 2025b; Kong et al., 2025; Janson et al., 2026; Han et al., 2024; Mo et al., 2025), and parameter-efficient adaptation (Hu et al., 2022; Zhang et al., 2023a; Dettmers et al., 2023; Kopiczko et al., 2023; Liu et al., 2024d; Meng et al., 2024; Wang et al., 2024; Li et al., 2024a)—provide complementary ways to reduce training costs. Operator-aware allocation adds a structural criterion: compression capacity can be assigned according to the functional role and measured redundancy of each component. This motivates a lightweight, reusable QK-specific design.

The missing QK efficiency paradigm. Attention naturally benefits from operator-aware design because its components serve distinct functional roles. While targeted efficiency mechanisms have matured for the Key-Value (KV) side (e.g., MQA (Shazeer, 2019), GQA (Ainslie et al., 2023), MLA-style compressed-KV attention (Liu et al., 2024a), and long-context caching (Xiao et al., 2023; Zhang et al., 2023b; Li et al., 2024b; Cai et al., 2024; Liu et al., 2024e; Tang et al., 2024; Yuan et al., 2025; DeepSeek-AI, 2026)) and the Value-Output (VO) side (Qiu et al., 2025), the QK path fundamentally lacks a unified, training-oriented paradigm. Existing QK explorations mostly target

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.

narrow objectives—such as head-sharing (Cordonnier et al., 2020; Javadi et al., 2023), KV-budget constraints (Yao et al., 2026; Hu et al., 2025), inference filtering (Godey et al., 2025; Li et al., 2025), or query-key normalization, QK-weight analysis, and explicit Q/K interaction (Henry et al., 2020; Chen & Yan, 2024; Gahtan & Bronstein, 2026)—without addressing the fundamental geometric organization of trained queries and keys.

This structural gap raises a central question:

> **Question.** *In modern LLMs, do queries and keys truly need to be learned as two independent full-size projections, or do trained models reveal a measurable shared routing geometry that can be exploited by a simple trainable QK module?*

To answer this question, we systematically investigate the **geometric properties of Q/K activations** and introduce a novel **operator-aware efficiency framework**. Our main contributions are summarized as follows:

(C1) Discovery of multi-anchor QK-coupling. We conduct a cross-scale empirical study and identify **multi-anchor QK-coupling** as a robust structural pattern in trained LLMs. Across the profiled model families and layers, queries and keys form coupled anchor-relative trajectories under multiple anchors, providing empirical evidence of **shared anchor-relative relational structure** that motivates an operator-aware QK parameterization.

(C2) Shared routing trunk with private residual branches. We formalize this observation into a **QK-coupled parameterization with a shared routing trunk and private residual branches**. Our score-space analysis characterizes the complementary supports and finite-rank approximation errors of query-private and key-private corrections around a fixed shared representation. It provides a structural criterion for allocating private capacity, with the resulting placement choices evaluated under matched budgets. The resulting defaults use query-private correction for standard attention and LoRA-style fine-tuning, and joint query-private/key-private correction for MLA-style attention; the placement choices are assessed by matched-budget pre-training ablations.

(C3) Efficient QK-block training across regimes. We instantiate the shared-private QK form across **standard pre-training, MLA-style compressed-KV attention pre-training, and LoRA-style fine-tuning**. Experiments validate the savings across these regimes: in standard-attention pre-training, QK-coupling reduces QK-operator FLOPs and optimizer-state memory by 75% while improving validation perplexity across 60M–1.3B models; in a DeepSeek-style 3B MLA run trained for 200B tokens, it achieves the best final test perplexity with a 50.57% QK-path projection-FLOP reduction. Latent KV-cache savings are demonstrated on the SmolLM-style MLA backbones, and the fine-tuning experiments use compact QK adapters.

A comprehensive review of **Related Work** is provided in Appendix B, and the notation table used throughout the paper is provided in Appendix A.

2 PRELIMINARIES

In this section, we briefly introduce the notation for the QK path, MLA-style compressed-KV attention, and low-rank/LoRA-style reparameterization used throughout the paper.

Multi-head attention and the QK path. Let $X_l \in \mathbb{R}^{T\times d_{\text{model}}}$ denote the input hidden states of the l-th Transformer layer, where T is the sequence length and d_{model} is the hidden dimension. For an equal-width attention path, the query, key, and value activations are

$$Q_l = X_l W_l^Q, \qquad K_l = X_l W_l^K, \qquad V_l = X_l W_l^V, \tag{1}$$

where $W_l^Q, W_l^K \in \mathbb{R}^{d_{\text{model}}\times d_{\text{qk}}}$ and $W_l^V \in \mathbb{R}^{d_{\text{model}}\times d_v}$. Thus $Q_l, K_l \in \mathbb{R}^{T\times d_{\text{qk}}}$ and $V_l \in \mathbb{R}^{T\times d_v}$. Let H denote the number of attention heads and let d_{head} denote the per-head Q/K dimension. The displayed equal-width notation uses $d_Q = d_K = d_{\text{qk}} = Hd_{\text{head}}$. For native GQA, we retain separate widths $d_Q = H_Q d_{\text{head}}$ and $d_K = H_{\text{KV}} d_{\text{head}}$; the corresponding parameter counts are given in Appendix F. For the i-th head, its corresponding attention can be computed as

$$\text{Attn}_{l,i}(X_l) = \text{Softmax}\Big(Q_{l,i}K_{l,i}^\top/\sqrt{d_{\text{head}}}\Big)\, V_{l,i}, \tag{2}$$

where $Q_{l,i}, K_{l,i} \in \mathbb{R}^{T\times d_{\text{head}}}$, and $V_{l,i}$ denotes the corresponding value slice. We refer to W_l^Q, W_l^K

and their activation outputs Q_l, K_l as the *QK path*. This path forms attention scores before values are aggregated, and therefore determines the addressing pattern of attention.

MLA-style compressed-KV attention. We use *MLA-style compressed-KV attention* for designs that reduce KV-cache cost by storing a compressed KV state instead of full keys and values. A simplified formulation is

$$C_l^{KV} = X_l W_l^{DKV}, \qquad K_l^C = C_l^{KV} W_l^{UK}, \qquad V_l = C_l^{KV} W_l^{UV}, \tag{3}$$

where $C_l^{KV} \in \mathbb{R}^{T\times d_c}$ is the compressed cached content latent, $K_l^C \in \mathbb{R}^{T\times d_{\text{qk}}}$, and $V_l \in \mathbb{R}^{T\times d_v}$. The query path can also be produced through a latent projection,

$$C_l^Q = X_l W_l^{DQ}, \qquad Q_l^C = C_l^Q W_l^{UQ}, \tag{4}$$

with $Q_l^C \in \mathbb{R}^{T\times d_{\text{qk}}}$. In practical MLA-style compressed-KV attention designs, the full score also contains positional branches:

$$Q_l = [Q_l^C, \mathrm{RoPE}(Q_l^R)], \qquad K_l = [K_l^C, \mathrm{RoPE}(K_l^R)]. \tag{5}$$

During decoding, the query branch is recomputed for the current token and is not cached, while the key path stores the content latent together with the positional key branch. Thus, the per-token cache size is proportional to $d_c + d_{\text{pos}}$, where d_{pos} is the RoPE/positional key-cache dimension. This notation will be used when comparing multi-anchor QK-coupling in standard attention and MLA-style compressed-KV attention.

LoRA-style adaptation. For LoRA-style fine-tuning, the pre-trained weight W_0 is frozen and only a low-rank update is learned:

$$W = W_0 + \Delta W, \qquad \Delta W = AB, \qquad A \in \mathbb{R}^{d_{\text{in}}\times r}, \quad B \in \mathbb{R}^{r\times d_{\text{out}}}, \tag{6}$$

with $r \ll \min(d_{\text{in}}, d_{\text{out}})$. We use this notation for low-rank Q/K reparameterization in pre-training and for Q/K low-rank updates in LoRA-style fine-tuning.

3 Key Observation: Multi-Anchor QK-coupling in Trained LLMs

Existing studies have revealed rich structural regularities in LLM activations, including low-rank activation structure (Cui et al., 2020; Huh et al., 2021; Yu & Wu, 2023; Liu et al., 2025b), cross-layer activation dependencies (Liu et al., 2024c; Kong et al., 2025; Brandon et al., 2024), and structured depth-wise representation geometry (Kriegeskorte et al., 2008; Raghu et al., 2017; Kornblith et al., 2019; Ethayarajh, 2019). These observations suggest that useful structure can be found in model activations, motivating us to examine a finer-grained object: the activation outputs of the Query and Key operators that form attention scores. We identify **multi-anchor QK-coupling**: Q and K exhibit aligned anchor-relative trajectories across depth while retaining operator-specific activations.

> **Key observation.** Trained LLMs exhibit aligned Q/K anchor-relative trajectories across depth while retaining operator-specific activations and activation scales.

3.1 Multi-Anchor QK-coupling Metric.

We compare how each operator family relates to the same anchor-layer index, using the operator-specific activations Q_a and K_a: for $A \in \{Q, K\}$ and anchor a, we collect Frobenius similarities from A_a to all non-anchor layers A_l, forming the anchor-relative trajectory below, as detailed in Appendix E.8.

$$\mathbf{u}_A^{(a)} := (\cos_F(A_l, A_a))_{1\le l\le L}^{l\neq a} \in \mathbb{R}^{L-1}, \quad \text{where } \cos_F(A,B) = \frac{\langle A, B\rangle_F}{\|A\|_F \|B\|_F}. \tag{7}$$

The anchor self-similarity is excluded, so the score is computed entirely from cross-layer relations. We define the multi-anchor QK-coupling score under anchor a as the cosine similarity between the two anchor-relative trajectories:

$$\kappa_{QK}(a) = \cos\left(\mathbf{u}_Q^{(a)}, \mathbf{u}_K^{(a)}\right). \tag{8}$$

Geometric intuition. Figure 1 illustrates how $\kappa_{QK}(a)$ measures directional agreement between the two anchor-relative trajectories. When $\kappa_{QK}(a) \approx 1$, the trajectories are approximately proportional, $\mathbf{u}_K^{(a)} \approx c_a \mathbf{u}_Q^{(a)}$ with $c_a > 0$. This shared relational organization motivates a routing parameterization with common coordinates and operator-specific readouts.

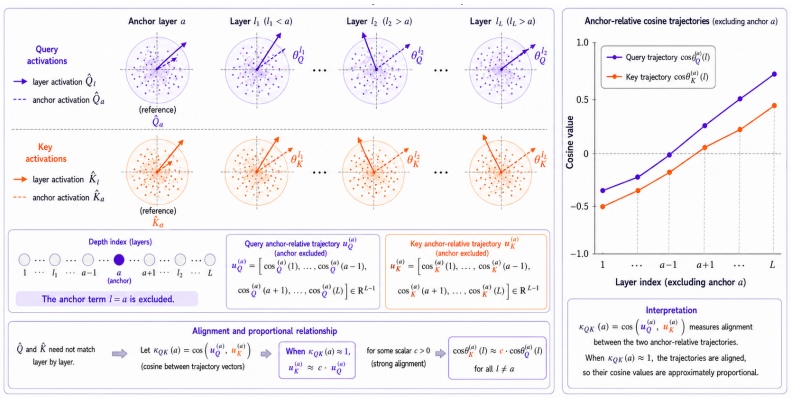

Figure 1: **Geometric intuition of multi-anchor QK-coupling.** For a fixed anchor-layer index a, a high $\kappa_{QK}(a)$ indicates directional agreement between the operator-specific anchor-relative trajectories. When $\kappa_{QK}(a) \approx 1$, these trajectories are approximately proportional. The anchor self-term is excluded.

3.2 Robustness Across Profiled Model Families and Training Emergence

Training-time emergence in a 15B-A2B MoE model with MLA-style compressed-KV attention. We first examine the training dynamics of multi-anchor QK-coupling using a 15B-A2B MoE model with a DeepSeek-V2-style implementation of MLA-style compressed-KV attention; detailed architecture and profiling settings are provided in Appendix C.1. Figure 2 shows the multi-anchor QK-coupling results across checkpoints as training tokens increase from 4.19B to 809.50B, where each cell reports $\kappa_{QK}(a)$ for one checkpoint and one anchor layer with the average multi-anchor QK-coupling score $\bar{\kappa}_{QK}$ increasing from 0.496 to 0.952. Crucially, multi-anchor QK-coupling appears across a broad range of anchor layers, suggesting the model organically learns this shared geometry over time.

Robustness across profiled model families. To examine multi-anchor QK-coupling across model families, we evaluate open-weight LLMs including LLaMA-3 8B (Grattafiori et al., 2024), Qwen2.5-3B Base (Team, 2024), and Qwen3-30B-A3B (Yang et al., 2025). For the 8B dense/GQA model and the 30B-A3B MoE/GQA model, we observe strong multi-anchor QK-coupling with $\bar{\kappa}_{QK} > 0.7$, supporting that this shared routing geometry is robust across the profiled model families. Also, Qwen2.5-3B Base has a mean score of $\bar{\kappa}_{QK} = 0.338$, its anchor-wise heatmap shows that many anchors remain above 0.3, suggesting a meaningful coupling pattern in this smaller base model. This corresponds to a 33.8% reduction in the mean squared distance between unit-normalized Q/K trajectories relative to orthogonality, indicating nontrivial positive relational alignment (Appendix E.8). Furthermore, ablation experiments on other operator pairs, including QV, KV, and VO, in the 15B-A2B MLA-style compressed-KV attention MoE model (Figure 2(b)) yield significantly lower multi-anchor QK-coupling scores, supporting that the phenomenon is specific to the QK addressing path.

3.3 From Multi-Anchor QK-coupling to Shared Routing Geometry

Sections 3.1–3.2 identify a robust anchor-relative QK-coupling signal across the profiled models. We now examine the shared low-dimensional routing structure suggested by this signal, using the following analyses to motivate a QK parameterization that preserves both common geometry and operator-specific flexibility (Appendix E.3).

Theoretical bridge: layer-wise Q/K geometry. To formalize this relationship, we define the layer-wise Q/K similarity matrices $G_Q, G_K \in \mathbb{R}^{L\times L}$:

$$(G_Q)_{\ell a} = \cos_F(Q_\ell, Q_a), \qquad (G_K)_{\ell a} = \cos_F(K_\ell, K_a), \qquad 1 \le \ell, a \le L. \tag{9}$$

The anchor-relative trajectories $\mathbf{u}_Q^{(a)}$ and $\mathbf{u}_K^{(a)}$ correspond to the off-diagonal columns of these matrices. The following theorem relates multi-anchor QK-coupling to the approximation error of a shared low-dimensional relational representation.

Theorem 1 (Informal). ***Multi-anchor QK-coupling supports shared low-dimensional routing geometry.*** *Suppose the normalized Q/K anchor-relative trajectories have an approximately*

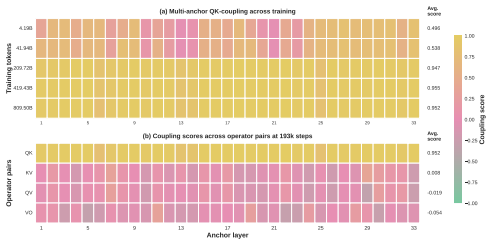

Figure 2: Multi-anchor QK-coupling emerges during training and comparisons for other operator pairs for 15B-A2B model.

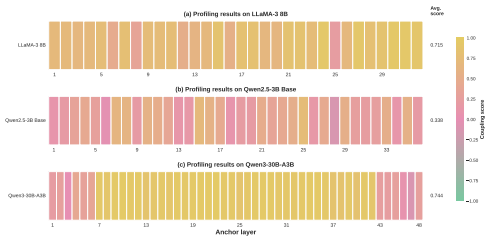

Figure 3: Multi-anchor QK-coupling emerges for additional models including LLaMA-3 8B, Qwen2.5-3B Base and Qwen3-30B-A3B.

> *low-rank common component. Then both trajectory families admit reconstruction in a shared low-dimensional subspace with separate coefficients. The reconstruction error is controlled by the common component's spectral tail and the mismatch measured by multi-anchor QK-coupling (formalized in Appendix D.1).*

Theorem 1 motivates our core parameterization: we use a shared low-rank channel projection to represent common routing geometry once, together with separate Q/K output maps that preserve the operators' asymmetric use of these shared coordinates.

MLA-style compressed-KV attention as an amplifier. Theorem 1 is architecture-agnostic: it links multi-anchor QK-coupling to shared routing geometry without assuming a specific attention implementation. Empirically, however, this coupling is most pronounced in our MLA-style compressed-KV attention model under a comparable activated-parameter budget. This motivates an MLA-specific explanation: a low-rank content representation can preserve common Q/K routing structure while suppressing operator-specific relational components. The following theorem formalizes this core-preserving mechanism and its effect on multi-anchor QK-coupling; see Appendix E.3 for how this specialization fits into the empirical-to-theoretical narrative.

> **Theorem 2** (Informal)**.** ***Coupling gain under common-core-preserving MLA-style compression.*** *Under the common-core separation conditions in Appendix D.2, suppose the joint KV core is spectrally dominant and the common and operator-specific relational components are mutually orthogonal. With queries fixed, optimal low-rank KV reconstruction at the core rank preserves the common structure and removes key-specific residuals. Coupling strictly increases whenever the removed relational residual is nonzero.*

Together, Theorems 1 and 2 motivate our shared-routing design: multi-anchor QK-coupling supports a common low-dimensional relational representation, while core-preserving MLA-style compression can strengthen this coupling. These motivate representing common Q/K routing coordinates once and retaining operator-specific flexibility through separate output maps and private residual branches.

4 QK-COUPLED PARAMETERIZATION FOR EFFICIENT LLMS

Building on the empirical QK-coupling signal and the shared-routing theory in Section 3.3, we now instantiate the resulting geometry as a trainable QK-coupled architecture.

4.1 CORE DESIGN: SHARED ROUTING TRUNK AND PRIVATE RESIDUAL BRANCHES

QK-coupling is **driven by an operator-specific signal**: trained Q/K activations exhibit similar anchor-relative relational organization across depth. We use this observation to motivate a shared routing trunk with operator-specific output maps and private residual branches, and evaluate the resulting quality–efficiency trade-off through matched-budget experiments.

We implement this principle by decomposing the QK addressing path into a **shared routing trunk** and lightweight **private residual branches**. The trunk captures the common routing geometry once, reducing QK projection cost, while query-private / key-private corrections preserve asymmetric score formation. This makes QK-coupling a budget-efficient, geometry-aware alternative to rank- or cache-driven compression; see Appendix E.1.

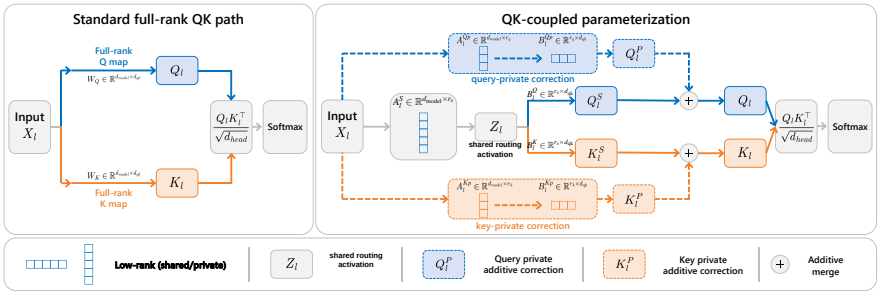

Figure 4: Illustration of the core design of QK-coupling architecture.

Core design 1: shared routing trunk. Motivated by Theorem 1, we replace the independent full-shape Q/K projections with a shared routing trunk. We introduce a shared routing matrix $A_l^S \in \mathbb{R}^{d_{\text{model}} \times r_s}$ ($r_s < d_{\text{model}}$), along with separate output maps $B_l^Q, B_l^K \in \mathbb{R}^{r_s \times d_{\text{qk}}}$:

Shared routing trunk with separate Q/K output maps:

$$Z_l = X_l A_l^S, \qquad Q_l^S = Z_l B_l^Q, \qquad K_l^S = Z_l B_l^K. \tag{10}$$

Here, $Z_l \in \mathbb{R}^{T \times r_s}$ serves as the routing activation, and $Q_l^S, K_l^S \in \mathbb{R}^{T \times d_{\text{qk}}}$ are sliced into H heads of Q/K width d_{head}. The separate output maps allow Q and K to utilize these shared routing coordinates asymmetrically.

Core design 2: private residual branches. To complement the shared Q/K routing trunk, we instantiate side-specific score-space residuals with low-rank query-private and key-private branches (see Appendix E.7 for detailed discussions for the necessity):

$$Q_l^P = X_l A_l^{Qp} B_l^{Qp}, \qquad K_l^P = X_l A_l^{Kp} B_l^{Kp}, \tag{11}$$

where $A_l^{Qp} \in \mathbb{R}^{d_{\text{model}} \times r_q}$, $B_l^{Qp} \in \mathbb{R}^{r_q \times d_{\text{qk}}}$, $A_l^{Kp} \in \mathbb{R}^{d_{\text{model}} \times r_k}$, and $B_l^{Kp} \in \mathbb{R}^{r_k \times d_{\text{qk}}}$. The head-specific tensors $Q_{l,g}^P$ and $K_{l,g}^P$ are obtained by slicing the total outputs along the Q/K channel dimension. In our practical implementation, the bottleneck dimensions r_s, r_q, and r_k are manually set fixed hyperparameters.

To integrate the private residual branches smoothly, we define mixing coefficients λ^S and λ^P using non-learnable switches $m_{l,g}^Q, m_{l,g}^K \in \{0, 1\}$ and learnable per-head scalars $\alpha_{l,g}^Q, \alpha_{l,g}^K \in (0, 1)$:

$$\lambda^S(m, \alpha) = \sqrt{1 - m(1 - \alpha)}, \qquad \lambda^P(m, \alpha) = \sqrt{m(1 - \alpha)}. \tag{12}$$

The final QK outputs are unified as:

$$Q_{l,g} = \lambda^S(m_{l,g}^Q, \alpha_{l,g}^Q) Q_{l,g}^S + \lambda^P(m_{l,g}^Q, \alpha_{l,g}^Q) Q_{l,g}^P, \tag{13a}$$

$$K_{l,g} = \lambda^S(m_{l,g}^K, \alpha_{l,g}^K) K_{l,g}^S + \lambda^P(m_{l,g}^K, \alpha_{l,g}^K) K_{l,g}^P. \tag{13b}$$

The square-root form gives squared branch coefficients α and $1 - \alpha$ when the private branch is enabled, providing an interpretable nominal energy allocation; Appendix E.2 gives the corresponding norm identity and its equal-energy, weak-correlation interpretation.

Figure 4 summarizes the core QK-coupling design: shared routing for common Q/K geometry, private branches for asymmetric score correction, and square-root mixing for controlled branch scaling.

4.2 Theory-Guided Private Correction Placement and Instantiations

Private-branch placement is a score-space reachability question: query-private correction moves queries against the shared key support, while key-private correction changes the key-side support against shared queries. Theorem 3 characterizes these complementary first-order corrections, with a plain-language explanation in Appendix E.6.

Theorem 3 (Informal)**.** ***Private residual branches recover score-space residuals.*** *Fix the shared routing trunk and consider first-order corrections to its score-space residuals:*

Table 1: Cost accounting for the QK-coupled parameterization across LLM training regimes. A concrete numerical instantiation of these formulas is provided in Appendix E.4.

Regime	Coupled path♣	Correction type	Saved item	Original cost♡	QK-coupled cost♡
Standard attention pre-training	X_l	Q	Params / Mem / FLOPs	$2d_{\text{model}}d_{\text{qk}}$	$d_{\text{model}}r_s + 2r_s d_{\text{qk}}$ $+d_{\text{model}}r_q + r_q d_{\text{qk}}$
MLA-style pre-training	trunk: C_l^S residual: X_l	Q, K	Params / Mem / FLOPs KV cache	$P_{\text{std}}^{\text{MLA}\diamond}$ $d_c + d_{\text{pos}}$	$P_{\text{qkc}}^{\text{MLA}\diamond}$ $d_s + d_{kp} + d'_{\text{pos}}$
LoRA-style fine-tuning	$\Delta W_l^Q, \Delta W_l^K$	Q	Params / Mem / FLOPs	$2r(d_{\text{model}} + d_{\text{qk}})$	$d_{\text{model}}r_s + 2r_s d_{\text{qk}}$ $+d_{\text{model}}r_q + r_q d_{\text{qk}}$

♣"Coupled path" specifies where the parameterization is applied. In MLA, C_l^S feeds $Z_l = C_l^S A_l^S$, while uncached X_l feeds the private branches.

♡ Costs are QK parameter counts; MLA also includes the value up-projection affected by the cached latent. Training-state storage is counted in parameter-proportional scalar elements: $4P$ for Adam (Kingma & Ba, 2014) (parameters, gradients, and two moments) and $3P$ for Muon (Jordan et al., 2024; Liu et al., 2025a) (parameters, gradients, and momentum), where P is the matrix-parameter count. Training projection FLOPs are $6NP$, with $N = BT$. These analytical counts are reported separately from measured GPU memory. KV-cache counts are in elements per token per layer.

♢ $P_{\text{std}}^{\text{MLA}}, P_{\text{qkc}}^{\text{MLA}}$ use the direct-query formulas in Appendix F.2; the compressed-query baseline uses (63).

1. ***Support-limited Recovery:*** *Query-private correction can only recover residuals that are right-supported in the shared key subspace, whereas key-private correction only recovers residuals left-supported in the shared query subspace.*
2. ***Synergistic Joint Correction:*** *The two branches open complementary score-space directions. Holding query-private capacity fixed, adding key-private capacity strictly reduces the optimal residual error when a nonzero residual component lies outside the query-reachable support but inside the key-reachable support.*

A formal orthogonal block decomposition and finite-rank error bounds are provided in Appendix D.3.

Theorem 3 provides a rank-budgeted score-space criterion that motivates query-private / key-private correction placement. The two correction modes make different score-space residual directions reachable by perturbing queries against the shared key representation and keys against the shared query representation, respectively. Thus, key-private correction complements query-private correction when residual components lie outside the query-reachable support but within the key-reachable support. The private ranks determine the approximation accuracy within these supports.

Guided by this analysis, we use query-private correction for standard attention and LoRA-style fine-tuning, which adds asymmetric score flexibility without the extra parameter and projection-FLOP overhead of key-private correction. We use joint query-private / key-private correction for MLA-style compressed-KV attention, where the cached low-rank latent creates an additional key-path bottleneck.

Typical regimes for QK-coupled training. We instantiate this framework across three typical regimes in the LLM training workflow.

Regime 1: Standard attention pre-training. We enable query-private correction and keep the key path inside the shared routing trunk:

$$K_{l,g} = K_{l,g}^S, \qquad Q_{l,g} = \lambda^S(m_{l,g}^Q, \alpha_{l,g}^Q)Q_{l,g}^S + \lambda^P(m_{l,g}^Q, \alpha_{l,g}^Q)Q_{l,g}^P. \tag{14}$$

Regime 2: MLA-style compressed-KV attention pre-training. The QK-coupled parameterization is applied to the content branches using the cached shared content latent $C_l^S \in \mathbb{R}^{T\times d_s}$. The optional key-private cached latent has dimension d_{kp}:

$$\begin{aligned} Z_l &= C_l^S A_l^S, \qquad Q_{l,g}^C = \lambda^S(m_{l,g}^Q, \alpha_{l,g}^Q)Q_{l,g}^{C,S} + \lambda^P(m_{l,g}^Q, \alpha_{l,g}^Q)Q_{l,g}^{C,P}, \\ K_{l,g}^C &= \lambda^S(m_{l,g}^K, \alpha_{l,g}^K)K_{l,g}^{C,S} + \lambda^P(m_{l,g}^K, \alpha_{l,g}^K)K_{l,g}^{C,P}. \end{aligned} \tag{15}$$

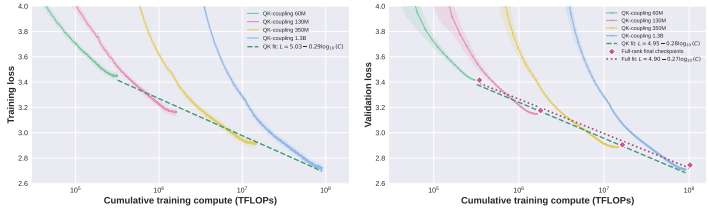

Figure 5: Scaling performance for standard attention training tasks for LLaMA-2-style models with model size scaling from 60M to 1.3B. The full-rank validation results are from reruns in our codebase under the same per-scale training and evaluation protocol.

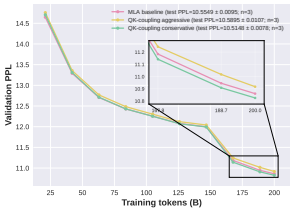

Figure 6: The 3B MLA-style pre-training comparison. The legend reports multi-run final-test summaries.

Table 2: Inference profiling of SmolLM-style MLA models: measured prefill latency, recompute-based decode throughput, peak allocated memory, and analytical latent KV-cache estimates.

Model	Variant	Prefill latency (s)	Recompute-based decode throughput (tok/s)	Peak alloc. (GiB)	Latent KV elems.	Latent KV saving
130M	MLA baseline	0.0761	13.586	0.354	56	0.00%
130M	QK-coupling	**0.0623**	**17.361**	**0.341**	**36**	**35.71%**
350M	MLA baseline	0.1183	7.854	0.836	72	0.00%
350M	QK-coupling	**0.0915**	**11.620**	**0.789**	**48**	**33.33%**

Regime 3: LoRA-style fine-tuning. The base weights are frozen, and the QK-coupled update applies solely to the trainable updates:

$$\Delta W_l^Q = \lambda^S(m_l^Q, \alpha_l^Q) A_l^S B_l^Q + \lambda^P(m_l^Q, \alpha_l^Q) A_l^{Qp} B_l^{Qp}, \qquad \Delta W_l^K = A_l^S B_l^K. \tag{16}$$

QK-coupled parameterization enables parameter-efficient QK training. The QK-coupled parameterization improves efficiency by using a shared routing trunk for Q and K, thereby reducing the trainable QK parameter proxy and the induced parameter storage and projection FLOPs. For standard attention with query-private correction, the original full Q/K parameter count is

$$P_{\text{std}}^{QK} = 2 d_{\text{model}} d_{\text{qk}}, \tag{17}$$

and the QK-coupled count is

$$P_{\text{qkc}}^{QK} = d_{\text{model}} r_s + 2 r_s d_{\text{qk}} + d_{\text{model}} r_q + r_q d_{\text{qk}}. \tag{18}$$

With a typical rank configuration $r_s = d_{\text{model}}/8$ and $r_q = d_{\text{model}}/16$ for standard attention pre-training, which is evaluated in our QK pre-training experiments, the QK-coupled parameterization achieves 75% savings in QK parameters, parameter storage, and per-step QK projection FLOPs compared with standard full-rank Q/K training under $d_{\text{qk}} = d_{\text{model}}$.

In the MLA-style setting, the baseline already compresses the content path into a cached latent, so QK-coupling targets the remaining redundancy inside this compressed addressing path. It replaces the baseline proxy $P_{\text{std}}^{\text{MLA}}$ with $P_{\text{qkc}}^{\text{MLA}}$, giving the same saving ratio $1 - P_{\text{qkc}}^{\text{MLA}}/P_{\text{std}}^{\text{MLA}}$ for parameters, Adam storage, and content-branch training FLOPs. The per-token KV cache changes from $d_c + d_{\text{pos}}$ to $d_s + d_{kp} + d'_{\text{pos}}$, so additional cache saving holds exactly when $d_s + d_{kp} + d'_{\text{pos}} < d_c + d_{\text{pos}}$. Full accounting is provided in Appendix F.

5 Experiments

In this section, we present a series of experiments to evaluate whether QK-coupling can reduce the QK addressing path while preserving training and adaptation quality. We test the three regimes in Section 4.2: standard-attention pre-training, MLA-style compressed-KV attention, and LoRA-style fine-tuning. We also present ablations on private-branch placement. Detailed setup and additional results are placed in Appendix C, including multi-seed runs, controls, rank allocations, and training-efficiency measurements (Appendix C.6).

Standard-attention pre-training. We first pre-train LLaMA-2-style models Touvron et al. (2023) from 60M to 1.3B on C4-en (Raffel et al., 2020) following the setup of Zhao et al. (2024). Figure 5

Table 3: Evaluation score on Qwen3.5-35B-A3B fine-tuning tasks. The parameter row reports theoretical non-frozen full-attention QK adapter parameters.

Dataset	Metric	QK-coupling (Ours)	ReLoRA	LoRA	ThinKeys
QK Params(M)		**0.61**	0.84	0.84	0.84
GSM8K	exact_match (↑)	**87.57**	83.49	87.49	87.34
BoolQ	accuracy (↑)	**91.47**	88.56	91.31	90.46
WikiText	word_perplexity (↓)	**8.303**	8.304	12.69	13.04
HumanEval	humaneval pass@1 (↑)	**40.24**	37.80	33.54	36.59
MMLU	accuracy (↑)	**83.39**	82.99	82.88	81.86

Table 4: Private-branch placement ablations under matched budgets. **Left**: Standard-attention pre-training. **Right**: MLA-style compressed-KV attention (original single-run comparison).

Variant	Model	r_q	r_k	PPL	Model	r_q	r_k	PPL
balanced	60M	16	16	30.37	350M	32	32	18.06
Q-private	60M	32	0	**30.29**	350M	64	0	**17.92**
K-private	60M	0	32	30.44	350M	0	64	18.03

Variant	r_q	d_{kp}	PPL
balanced	64	64	**10.5827**
Q-private	128	0	10.7193
K-private	0	128	10.6736

shows that QK-coupling preserves stable scaling and matches or improves the full-rank validation perplexity across all model sizes, while reducing QK-operator FLOPs and optimizer-state memory by 75%. Detailed configurations and per-scale numbers are provided in Tables 8 and 9.

MLA-style compressed-KV attention. We next evaluate the compressed-KV setting on DeepSeek-style (Liu et al., 2024a) pre-training. In the 3B MoE setting, trained for 200B tokens with sequence length $T = 4096$, Figure 6 reports validation trajectories and multi-seed final-test summaries. Conservative QK-coupling outperforms the baseline while reducing the QK-path projection-FLOP proxy by 50.57%. The aggressive configuration increases this saving to 53.44% with mean test PPL a 0.33% increase over the baseline mean. Detailed statistics appear in Appendix C.6.1; the 9B run and full setup are reported in Appendix C.4.

On SmolLM-style (Allal et al., 2025) MLA backbones, we focus on inference-side efficiency. Table 2 shows that QK-coupling reduces the latent KV-cache estimate by 35.71%/33.33% on 130M/350M models, while also improving prefill latency, recompute-based decode throughput, and peak allocated memory. The corresponding SmolLM pre-training results are deferred to Appendix C.3.

LoRA-style fine-tuning. We evaluate QK-coupling as a low-rank Q/K adapter. On Qwen3.5-35B-A3B (Qwen Team, 2026), Table 3 shows that QK-coupling uses fewer trainable QK adapter parameters than ReLoRA (Lialin et al., 2023)/LoRA (Hu et al., 2022)/ThinKeys (Yao et al., 2026) (0.61M vs. 0.84M) and obtains the best or tied-best result on all reported task groups. Encoder-style GLUE fine-tuning and the full MMLU breakdown are provided in Appendix C.5 and Table 19.

Private-branch placement ablation. Table 4 examines the correction placement motivated by Theorem 3 under matched private budgets. Standard attention favors query-private correction on both 60M and 350M models, whereas MLA-style attention favors balanced query-private / key-private correction on the 3B run. This is consistent with the score-space analysis of complementary query-side and key-side corrections; details are in Appendices C.2.3 and C.4.2.

6 Conclusion and Future Work

We introduced multi-anchor QK-coupling and a shared-routing parameterization with private residual branches. Relational and score-space analyses motivate the design, while experiments across standard pre-training, MLA-style pre-training, and LoRA-style fine-tuning demonstrate favorable quality–efficiency trade-offs. Multi-seed results, matched-budget controls, and measured training efficiency support the approach. Future work includes automatic rank allocation, broader architectural evaluation, and hardware-aware kernels.

REPRODUCIBILITY STATEMENT

Our model configurations, rank settings, and training and evaluation protocols are detailed in Appendix C, with multi-seed results, additional controls, and measured training efficiency in Appendix C.6. The assumptions and complete proofs of our theoretical results are provided in Appendix D. Appendix F details the parameter, storage, projection-FLOP, and KV-cache accounting used in our efficiency comparisons.

AI USE STATEMENT

Generative AI tools were used for manuscript structuring, language editing, LaTeX organization, and feedback on the experimental evaluation protocol. They also assisted the revision and mathematical cross-checking of the theoretical derivations and proofs in this paper. No experimental measurements were generated, estimated, or imputed during this revision. The authors are responsible for independently validating the final mathematics, citations, experiments, and disclosure before submission.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, 2023.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. Smollm2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

William Brandon, Mayank Mishra, Aniruddha Nrusimha, Rameswar Panda, and Jonathan Ragan-Kelley. Reducing transformer key-value cache size with cross-layer attention. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, et al. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Yiming Chen, Yuan Zhang, Yin Liu, Kun Yuan, and Zaiwen Wen. A memory efficient randomized subspace optimization method for training large language models. *arXiv preprint arXiv:2502.07222*, 2025.

Yiting Chen and Junchi Yan. What rotary position embedding can tell us: Identifying query and key weights corresponding to basic syntactic or high-level semantic information. *Advances in Neural Information Processing Systems*, 37:54507–54528, 2024.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*, 2019.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. Multi-head attention: Collaborate instead of concatenate. *arXiv preprint arXiv:2006.16362*, 2020.

Chunfeng Cui, Kaiqi Zhang, Talgat Daulbaev, Julia Gusak, Ivan Oseledets, and Zheng Zhang. Active subspace of neural networks: Structural analysis and universal attacks. *SIAM Journal on Mathematics of Data Science*, 2(4):1096–1122, 2020.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International conference on machine learning*, pp. 7480–7512. PMLR, 2023.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 8-bit optimizers via block-wise quantization. *arXiv preprint arXiv:2110.02861*, 2021.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 55–65, 2019.

Barak Gahtan and Alex M Bronstein. Coupled query-key dynamics for attention. *arXiv preprint arXiv:2604.01683*, 2026.

Nathan Godey, Alessio Devoto, Yu Zhao, Simone Scardapane, Pasquale Minervini, Éric de la Clergerie, and Benoît Sagot. Q-filters: Leveraging qk geometry for efficient kv cache compression. *arXiv preprint arXiv:2503.02812*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Andi Han, Jiaxiang Li, Wei Huang, Mingyi Hong, Akiko Takeda, Pratik Jawanpuria, and Bamdev Mishra. Sltrain: a sparse plus low rank approach for parameter and memory efficient pretraining. *Advances in Neural Information Processing Systems*, 37:118267–118295, 2024.

Yongchang Hao, Yanshuai Cao, and Lili Mou. Flora: Low-rank adapters are secretly gradient compressors. *arXiv preprint arXiv:2402.03293*, 2024.

Yutong He, Pengrui Li, Yipeng Hu, Chuyan Chen, and Kun Yuan. Subspace optimization for large language models with convergence guarantees. *arXiv preprint arXiv:2410.11289*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4246–4253, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, DDL Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 10, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Jingcheng Hu, Houyi Li, Yinmin Zhang, Zili Wang, Shuigeng Zhou, Xiangyu Zhang, and Heung Yeung Shum. Multi-matrix factorization attention. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25114–25126, 2025.

Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks. *arXiv preprint arXiv:2103.10427*, 2021.

Paul Janson, Edouard Oyallon, and Eugene Belilovsky. Stabilizing native low-rank llm pretraining. *arXiv preprint arXiv:2602.12429*, 2026.

Farnoosh Javadi, Walid Ahmed, Habib Hajimolahoseini, Foozhan Ataiefard, Mohammad Hassanpour, Saina Asani, Austin Wen, Omar Mohamed Awad, Kangling Liu, and Yang Liu. Gqkva: Efficient pre-training of transformers by grouping queries, keys, and values. *arXiv preprint arXiv:2311.03426*, 2023.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. *URL https://kellerjordan. github. io/posts/muon*, 6(3):4, 2024.

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

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Boao Kong, Junzhu Liang, Yuxi Liu, Renjia Deng, and Kun Yuan. Cr-net: Scaling parameter-efficient training with cross-layer low-rank structure. *arXiv preprint arXiv:2509.18993*, 2025.

Dawid J Kopiczko, Tijmen Blankevoort, and Yuki M Asano. Vera: Vector-based random matrix adaptation. *arXiv preprint arXiv:2310.11454*, 2023.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019.

Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:249, 2008.

Tao Li, Zhengbao He, Yujun Li, Yasheng Wang, Lifeng Shang, and Xiaolin Huang. Flat-lora: Low-rank adaptation over a flat loss landscape. *arXiv preprint arXiv:2409.14396*, 2024a.

Tenghui Li, Guoxu Zhou, Xuyang Zhao, Yuning Qiu, and Qibin Zhao. Efficient low rank attention for long-context inference in large language models. *arXiv preprint arXiv:2510.23649*, 2025.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024b.

Vladislav Lialin, Namrata Shivagunde, Sherin Muckatira, and Anna Rumshisky. Relora: High-rank training through low-rank updates. *arXiv preprint arXiv:2307.05695*, 2023.

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024a.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024b.

Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Reza Haffari, and Bohan Zhuang. Minicache: Kv cache compression in depth dimension for large language models. *Advances in Neural Information Processing Systems*, 37:139997–140031, 2024c.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025a.

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 *Forty-first International Conference on Machine Learning*, 2024d.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *arXiv preprint arXiv:2402.02750*, 2024e.

Ziyue Liu, Ruijie Zhang, Zhengyang Wang, Mingsong Yan, Zi Yang, Paul D Hovland, Bogdan Nicolae, Franck Cappello, Sui Tang, and Zheng Zhang. Cola: Compute-efficient pre-training of llms via low-rank activation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4627–4645, 2025b.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. *Advances in Neural Information Processing Systems*, 37: 121038–121072, 2024.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Zhanfeng Mo, Long-Kai Huang, and Sinno Jialin Pan. Parameter and memory efficient pretraining via low-rank riemannian optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. *Advances in neural information processing systems*, 31, 2018.

Matteo Pagliardini, Amirkeivan Mohtashami, Francois Fleuret, and Martin Jaggi. Denseformer: Enhancing information flow in transformers via depth weighted averaging. *Advances in neural information processing systems*, 37:136479–136508, 2024.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, et al. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. *arXiv preprint arXiv:2505.06708*, 2025.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

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

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: international conference for high performance computing, networking, storage and analysis*, pp. 1–16. IEEE, 2020.

Thomas Robert, Mher Safaryan, Ionut-Vlad Modoranu, and Dan Alistarh. Ldadam: Adaptive optimization from low-dimensional gradient statistics. *arXiv preprint arXiv:2410.16103*, 2024.

Noam Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International conference on machine learning*, pp. 4596–4604. PMLR, 2018.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context llm inference. *arXiv preprint arXiv:2406.10774*, 2024.

Kimi Team, Guangyu Chen, Yu Zhang, Jianlin Su, Weixin Xu, Siyuan Pan, Yaoyu Wang, Yucheng Wang, Guanduo Chen, Bohong Yin, et al. Attention residuals. *arXiv preprint arXiv:2603.15031*, 2026.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL `https://qwenlm.github.io/blog/qwen2.5/`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pp. 353–355, 2018.

Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. Lora-pro: Are low-rank adapters properly optimized? *arXiv preprint arXiv:2407.18242*, 2024.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, Damai Dai, Huazuo Gao, Jiang Chang, Kuai Yu, et al. mhc: Manifold-constrained hyper-connections. *arXiv preprint arXiv:2512.24880*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Hengshuai Yao, Xing Chen, Ahmed Murtadha, and Guan Wang. Thin keys, full values: Reducing kv cache via low-dimensional attention selection. *arXiv preprint arXiv:2603.04427*, 2026.

Hao Yu and Jianxin Wu. Compressing transformers: features are low-rank, but weights are not! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11007–11015, 2023.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Yuxing Wei, Lean Wang, Zhiping Xiao, et al. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23078–23097, 2025.

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. *arXiv preprint arXiv:2303.10512*, 2023a.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023b.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*, 2024.

Zhanchao Zhou, Tianyi Wu, Zhiyun Jiang, Fares Obeid, and Zhenzhong Lan. Value residual learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28341–28356, 2025.

Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. Hyper-connections. *arXiv preprint arXiv:2409.19606*, 2024.

Hanqing Zhu, Zhenyu Zhang, Wenyan Cong, Xi Liu, Sem Park, Vikas Chandra, Bo Long, David Z Pan, Zhangyang Wang, and Jinwon Lee. Apollo: Sgd-like memory, adamw-level performance. *Proceedings of Machine Learning and Systems*, 7, 2025.

Appendix

A Notation

Table 5 summarizes the main symbols used throughout the paper. A primed symbol, such as d'_{pos}, denotes the corresponding dimension after a QK-coupled dimension change.

Table 5: Notation used in the paper.

Symbol	Meaning
d_{model}	Hidden dimension.
H	Number of attention heads.
d_{head}	Per-head Q/K dimension.
d_{qk}	Total Q/K output dimension, usually Hd_{head}.
d_v	Total value dimension.
d_c	Baseline MLA-style compressed-KV attention cached content latent dimension.
d_s	QK-coupled variant of MLA-style compressed-KV attention shared cached latent dimension.
d_{kp}	Key-private cached latent dimension.
d_{pos}	RoPE / positional key-cache dimension.
r_s	Shared routing rank.
r_q, r_k	Query-private / key-private rank.
L	Number of layers.
T	Sequence length.
B	Batch size.
$N = BT$	Number of training tokens in a batch.

Terminology conventions. Throughout the paper, *multi-anchor QK-coupling* denotes the empirical Q/K geometry measured by κ_{QK}, and *anchor-relative trajectory* denotes vectors such as $\mathbf{u}_Q^{(a)}$ and $\mathbf{u}_K^{(a)}$. The trainable QK module consists of a *shared routing trunk* and optional *private residual branches*; side-specific additions are referred to as *query-private correction* and *key-private correction*. Residual score matrices are referred to as *score-space residuals*. We use *MLA-style compressed-KV attention* for designs that store compressed KV latents.

B Related Work

QK-path efficient attention. The QK path has been studied from several angles. Head-sharing and grouping methods reduce redundancy across attention heads, for example by sharing query/key projections or grouping queries, keys, and values (Cordonnier et al., 2020; Javadi et al., 2023). Cache- and capacity-driven methods modify the QK circuit under memory constraints: thin-key attention exploits the asymmetry between low-dimensional attention selection and high-dimensional value transfer, while Multi-matrix Factorization Attention (MFA) factorizes the QK circuit under strict KV-cache budgets (Yao et al., 2026; Hu et al., 2025). Inference-oriented methods use QK geometry or low-rank proxy scores for long-context filtering (Godey et al., 2025; Li et al., 2025). Other works control the QK path for logit stability, fine-tuning selection, or explicit QK interaction (Henry et al., 2020; Dehghani et al., 2023; Chen & Yan, 2024; Gahtan & Bronstein, 2026). These methods establish QK as an important efficiency target through head sharing, cache management, retrieval, stability, and adaptation. Our approach contributes an empirical multi-anchor QK diagnostic and a corresponding shared–private parameterization for efficient training.

Low-rank and efficient LLM training. Recent works improve LLM training efficiency by exploiting redundancy outside a specific attention operator. Optimizer- and gradient-side methods reduce the

memory cost of full-parameter training through optimizer-state partitioning, compressed adaptive statistics, or low-rank gradient representations (Rajbhandari et al., 2020; Shazeer & Stern, 2018; Dettmers et al., 2021; Zhao et al., 2024; He et al., 2024; Chen et al., 2025; Robert et al., 2024; Zhu et al., 2025). Architecture-level methods reduce computation or memory through low-rank activations, compact layer transformations, cross-layer low-rank residuals, or factorized weights (Hao et al., 2024; Liu et al., 2025b; Kong et al., 2025; Janson et al., 2026; Han et al., 2024; Mo et al., 2025). Parameter-efficient adaptation methods, including LoRA variants, restrict fine-tuning updates to structured low-dimensional subspaces (Hu et al., 2022; Zhang et al., 2023a; Dettmers et al., 2023; Kopiczko et al., 2023; Liu et al., 2024d; Meng et al., 2024; Wang et al., 2024; Li et al., 2024a). These methods are complementary to ours: they target global training states, broad model components, or general adaptation subspaces, while our approach targets how trained models organize the QK path through multi-anchor QK-coupling. The low-dimensional components in our method implement a parameterization guided by the measured multi-anchor QK-coupling signal.

Operator-aware attention and depth-wise information flow beyond QK. Efficient attention methods often exploit the functional role of a specific attention component. On the Key-Value (KV) side, Multi-Query Attention (MQA), Grouped-Query Attention (GQA), MLA-style compressed-KV attention, and long-context KV-compression methods reduce decoding cache or memory traffic (Shazeer, 2019; Ainslie et al., 2023; Liu et al., 2024a; Xiao et al., 2023; Zhang et al., 2023b; Li et al., 2024b; Cai et al., 2024; Liu et al., 2024e; Tang et al., 2024; Yuan et al., 2025; DeepSeek-AI, 2026). Recent systems further combine sparse KV compression with MLA-style compressed-KV attention for million-token contexts (DeepSeek-AI, 2026). On the Value-Output (VO) or value path, Gated Attention identifies output-side gating as an effective local modification inside softmax attention (Qiu et al., 2025), while value residual and sharing mechanisms propagate value information across layers (Zhou et al., 2025). More broadly, DenseFormer, Hyper-Connections (HC), Manifold-Constrained Hyper-Connections (mHC), and residual-attention methods improve depth-wise information flow by reweighting, expanding, constraining, or selectively aggregating residual representations across layers (Pagliardini et al., 2024; Zhu et al., 2024; Xie et al., 2025; Team et al., 2026). These works motivate local or depth-aware designs for the KV cache, value/output path, and residual stream. Complementing these directions, multi-anchor QK-coupling studies the joint relational organization of trained Q/K activations and uses it to guide a training-oriented QK parameterization.

Representation similarity and QK profiling. Our observation is related to representation similarity analysis, including RSA, SVCCA, PWCCA, and CKA (Kriegeskorte et al., 2008; Raghu et al., 2017; Morcos et al., 2018; Kornblith et al., 2019), which compare neural representations across layers, models, and training stages. Prior work also shows that Transformer layers can have structured depth-wise similarity and anisotropy (Ethayarajh, 2019). We assess operator specificity by comparing QK with the QV, KV, and VO controls under the same profiling protocol in Figure 2. We compare the anchor-relative trajectories induced by the query and key operators inside attention, excluding the trivial self-anchor match, and ask whether Q and K share the same routing geometry across depth. The profiling results are used to motivate a trainable QK-coupled parameterization.

C Experimental Details and Additional Experimental Results

C.1 Profiling Setup and Additional Results for Multi-anchor QK-coupling

In this subsection, we present the detailed setup and additional results for the multi-anchor QK-coupling profiling experiments reported in Section 3.

C.1.1 Profiling on a 15B-A2B Model with MLA-style Compressed-KV Attention

To examine the training-time formation of multi-anchor QK-coupling, we profile a 15B-A2B MoE Transformer built on a DeepSeek-V2-style implementation of MLA-style compressed-KV attention backbone (Liu et al., 2024a). Profiling intermediate pre-training checkpoints tracks the formation of anchor-relative Q/K relationships throughout training.

For each checkpoint, we run the model in evaluation mode and collect the attention-side activation tensors from all Transformer layers with sequence length 4096 and profiling batch size 8. We average the activations over the batch dimension and then compute the multi-anchor QK-coupling score defined in (8) for every anchor layer. Each heatmap entry corresponds to one anchor layer, and

Table 6: Configuration of the 15B-A2B MoE model with DeepSeek-V2-style implementation of MLA-style compressed-KV attention used for multi-anchor QK-coupling profiling. Layer indices are zero-based.

Category	Hyperparameter	Value
Model scale	Number of layers L	33
	Hidden dimension d_{model}	1536
	Padded vocabulary size	153376
MLA-style compressed-KV attention	Number of attention heads H	48
	Query latent rank	384
	Cached content latent dimension d_c	512
	Per-head content QK dimension d_{head}	128
	RoPE key-cache dimension d_{pos}	64
	Value head dimension	128
FFN / MoE	Dense FFN layers	0–1
	MoE layers	2–32
	FFN hidden size	7680
	Activation function	SwiGLU
	Number of experts per MoE layer	128
	Activated experts per token	8
	Routed expert hidden size	768
	Shared expert hidden size	1536
Pre-training	Sequence length T	4096
	Pre-training global batch size B	1024
	Tokens per training step	4.19M
	Profiled layers	33
	Profiled anchors	33

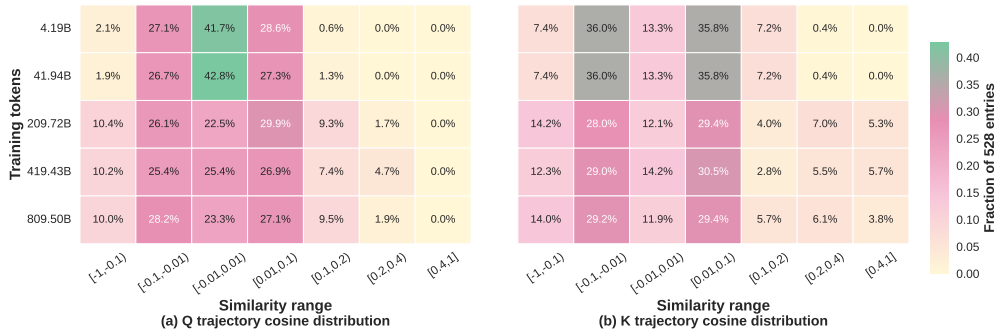

Figure 7: Distribution of anchor-relative similarities for query and key activations.

the reported average score is the mean over all 33 anchors. Training tokens are obtained from the checkpoint step multiplied by the pre-training global batch size and sequence length. To verify that the observed multi-anchor QK-coupling is specific to the query-key addressing path, we additionally profile several operator pairs at the final checkpoint. In this comparison, QK, KV, QV, and VO are evaluated under the same anchor selection, normalization, and averaging protocol. This control separates multi-anchor QK-coupling from generic depth-wise similarity that may appear in large Transformer blocks. The final multi-anchor QK-coupling score and the operator-pair ablation result are provided in Figure 2.

Distribution of anchor-relative similarities. We inspect the off-diagonal similarities underlying the anchor-relative trajectories. For each checkpoint and operator $A \in \{Q, K\}$, symmetry gives $\binom{33}{2} = 528$ distinct unordered layer-pair similarities $\cos_F(A_l, A_a)$ with $1 \le l < a \le L$. Figure 7 groups these values into coarse ranges. Q has zero mass in $[0.4, 1]$ at all reported checkpoints, while

Table 7: Model configurations used for open-weight multi-anchor QK-coupling profiling. The second block reports additional architecture and profiling-related dimensions. "0" denotes dense models without routed experts. For the Qwen3-30B-A3B model, the parameter count is reported as total/activated parameters per token.

Model	**Params (Billion)**	d_{model}	**Intermediate**	H	**KV Heads**	L	**Profile** T
LLaMA-3 8B	8.0	4096	14336	32	8	32	4096
Qwen2.5-3B Base	3.1	2048	11008	16	2	36	4096
Qwen3-30B-A3B	30.5/3.3	2048	6144	32	4	48	4096
Model	**Vocab**	**Max Pos.**	d_{head}	**Experts**	**Top-**k	**MoE Inter.**	
LLaMA-3 8B	128256	8192	128	0	0	0	
Qwen2.5-3B Base	151936	32768	128	0	0	0	
Qwen3-30B-A3B	151936	40960	128	128	8	768	

Table 8: Hyperparameter configurations for LLaMA-2 models of different scales.

Model	d_{model}	r_s	r_q	**Intermediate**	H	L	**Steps**	**Tokens (Billion)**	**Max lr**
60M	512	64	32	1376	8	8	10000	1.3	0.0025
130M	768	96	48	2048	12	12	20000	2.6	0.0010
350M	1024	128	64	2736	16	24	60000	7.8	0.0006
1.3B	2048	256	128	5461	32	24	100000	13.1	0.0006

K has at most 5.7% in this range. This distributional summary complements the anchor-wise coupling measurements in Figure 2.

C.1.2 Profiling on Other Models

To examine multi-anchor QK-coupling across public LLM families, we additionally profile three open-weight base models: LLaMA-3 8B (Grattafiori et al., 2024), Qwen2.5-3B Base (Team, 2024), and Qwen3-30B-A3B (Yang et al., 2025). LLaMA-3 8B and Qwen2.5-3B are dense decoder-only Transformers with grouped-query attention, while Qwen3-30B-A3B is a sparse MoE decoder with grouped-query attention (GQA) (Ainslie et al., 2023). These models cover different parameter scales, depths, vocabulary sizes, and dense/MoE feed-forward designs, supporting the cross-family evaluation of multi-anchor QK-coupling.

For each model, we use the released checkpoint. The model is run in evaluation mode, and we collect the activation tensors after the query and key projections from every Transformer block. For GQA models, query and key activations retain their native projected widths. Eq. (8) compares Q-to-Q and K-to-K anchor-relative trajectories before computing their coupling. For a model with L layers, we compute $\kappa_{QK}(a)$ for all L anchors and report both the anchor-wise heatmap and the mean score over anchors.

C.2 Experimental Details for C4 Pre-Training Tasks

C.2.1 Basic Experimental Setup

To evaluate QK-coupling in a standard-attention setting, we conduct LLaMA-2-style Touvron et al. (2023) pre-training with model scales ranging from 60M to 1.3B on the C4-en dataset (Raffel et al., 2020). We follow the experimental setup of Zhao et al. (2024), a widely used pre-training benchmark for efficient LLMs. We set the sequence length to $T = 256$ and the global batch size to $B = 512$. The learning-rate schedule uses two phases: an initial linear warm-up during the first 10% of training iterations, followed by cosine decay that gradually reduces the learning rate to 10% of its initial value. Complete architectural configurations and training protocol details are documented in Table 8.

Table 9: Scaling performance on C4-en pre-training. Both the full-rank baseline and the QK-coupled model are run in our codebase under the same per-scale training and evaluation protocol. "QK-coupled" denotes the QK-coupled parameterization.

Model	Full Params(M)	QK-coupled Params(M)	Param Saving(%)	Full TFLOPs/iter	QK-coupled TFLOPs/iter	Full PPL	QK-coupled PPL
60M	58.06	54.92	5.42	34.43	31.95	30.44	**30.29**
130M	134.09	123.47	7.92	89.83	81.48	23.92	**23.27**
350M	367.92	330.17	10.26	273.47	243.78	18.26	**17.92**
1.3B	1338.98	1187.99	11.28	1021.27	902.52	15.56	**15.00**

Table 10: Hyperparameter configurations and validation performance for query-private / key-private correction placement ablations.

Model	d_{model}	r_s	**Variant**	r_q	r_k	$r_q + r_k$	**Validation PPL**
60M	512	64	query-private	32	0	32	**30.29**
60M	512	64	key-private	0	32	32	30.44
60M	512	64	balanced	16	16	32	30.37
350M	1024	128	query-private	64	0	64	**17.92**
350M	1024	128	key-private	0	64	64	18.03
350M	1024	128	balanced	32	32	64	18.06

C.2.2 Scaling Performance in Pre-Training Tasks.

Table 9 shows that QK-coupling achieves stable pre-training scaling and comparable or better validation performance than the full-rank baseline, while reducing QK-operator FLOPs and optimizer-state memory by 75%. The table further summarizes the model-level scaling behavior under the standard-attention GaLore pre-training setup. With the embedding, LM head, FFN, and V/O projections held fixed, QK-path compression reduces total trainable parameters and per-iteration TFLOPs across all model scales, as reported in the table. These savings are accompanied by improved model quality: QK-coupling achieves lower validation perplexity than the full-rank baseline at the reported checkpoints for 60M, 130M, 350M, and 1.3B models.

C.2.3 Ablation on Query-Private / Key-Private Correction Placement

To further understand how to place query-private / key-private correction in the standard-attention regime, we conduct a controlled ablation on the 60M and 350M models under the same pre-training task. The shared routing rank is fixed as $r_s = d_{\text{model}}/8$ for all variants, and the total private-rank budget is fixed as $d_{\text{model}}/16$. The detailed hyperparameter configurations and validation results are provided in Table 10, which shows that query-private correction is more suitable for these standard-attention tasks and supports our theoretical clarification in Theorem 3.

C.2.4 Comparison on QK-efficient Baselines

We further compare QK-coupling with several QK-efficient baselines on the 350M standard-attention LLaMA-2-style model. This ablation uses the C4-en pre-training task in Appendix C.2.3, with global batch size $B = 768$ and a 40K-step training horizon. Full-rank QK, Grouped QK, and QK-coupling use learning rate 1.5×10^{-3}; the tuned Independent-LR QK control uses 1.0×10^{-3}, as reported in Table 11. We compare four QK parameterizations: full-rank QK, grouped QK, independent low-rank QK, and QK-coupling. Grouped QK follows the head-sharing and grouping principle used in efficient attention variants (Ainslie et al., 2023; Cordonnier et al., 2020), while independent low-rank QK applies separate low-rank factorizations to the Q and K projections, similar in spirit to standard low-rank adaptation (Hu et al., 2022). For a fair QK-budget comparison, QK-coupling uses $r_s = d_{\text{model}}/4$ and $r_q = d_{\text{model}}/8$, grouped QK uses 8 QK groups for 16 attention heads, and independent low-rank QK uses factorization rank $r = d_{\text{model}}/4$. The QK parameter counts, per-step QK projection FLOPs, and validation perplexities are reported in Table 11. Under the matched QK budget, QK-coupling achieves the best validation perplexity among the efficient QK variants and also outperforms the full-rank QK baseline at the reported checkpoints.

Table 11: QK-efficient parameterization comparison on the 350M model. Independent-LR QK uses the tuned learning rate 1.0×10^{-3}; all other methods use 1.5×10^{-3}. Unstable results under the common learning rate and the learning-rate sweep are reported in Appendix C.6.2.

Variant	QK Params(M)	QK TFLOPs/iter	Saving(%)	PPL@10K	PPL@20K	PPL@40K
Full-rank QK	50.33	59.37	0.00	23.12	19.79	18.05
Grouped QK	25.17	29.69	50.00	24.08	20.37	18.64
Independent-LR QK (tuned, LR $= 1.0 \times 10^{-3}$)	25.17	29.69	50.00	23.93	19.55	18.19
QK-coupling	25.17	29.69	50.00	**22.55**	**19.34**	**17.64**

Table 12: MLA-style compressed-KV attention QK parameter and FLOP comparison on SmolLM backbones.

Model	Variant	KV heads H_{KV}	Config.[a]	QK Params/layer(M)	QK Params(M)	QK TFLOPs/iter	Saving(%)
130M	MLA	3	$d_c = 32,\ d_{\mathrm{R,head}} = 8$	0.369	11.08	13.07	0.00
130M	QK-coupled	3	$r_s = 16,\ r_q = 8,\ r_k = 8,\ d'_{\mathrm{R,head}} = 4$	0.043	1.29	1.52	88.38
350M	MLA	5	$d_c = 32,\ d_{\mathrm{R,head}} = 8$	1.000	31.99	37.74	0.00
350M	QK-coupled	5	$r_s = 20,\ r_q = 8,\ r_k = 8,\ d'_{\mathrm{R,head}} = 4$	0.088	2.81	3.32	91.21

[a] $d_{\mathrm{R,head}}$ and $d'_{\mathrm{R,head}}$ are per-KV-head RoPE widths, giving baseline latent-cache estimates of $32 + 3 \times 8 = 56$ and $32 + 5 \times 8 = 72$ elements per token per layer in Table 2.

C.3 MLA-Style Compressed-KV Attention Pre-Training and Inference on SmolLM Backbones

We further evaluate QK-coupling in the MLA-style compressed-KV attention pre-training and inference regime using SmolLM-style (Allal et al., 2025) backbones modified with MLA-style compressed-KV attention (Liu et al., 2024a). We consider two model scales, 130M and 350M, and compare the standard MLA-style compressed-KV attention baseline with the QK-coupled variant. Both models are trained on the C4-en pre-training task (Raffel et al., 2020). The 130M model is trained for 3B tokens, and the 350M model is trained for 9B tokens. Within each scale, both variants use the same dataset and training recipe; the architectural differences comprise the shared/private content-path parameterization and the positional-key dimension specified in Table 12.

Pre-training performance. Table 13 shows that QK-coupling delivers strong pre-training quality together with substantial QK-path efficiency gains. On the 130M model, QK-coupling reaches a lower final validation perplexity at 3B tokens than the MLA-style compressed-KV attention baseline (30.23 vs. 30.45), while reducing QK-path parameters and projection FLOPs by 88.38%. On the 350M model, QK-coupling achieves lower validation perplexity at every reported checkpoint and improves the final perplexity from 23.49 to 23.36 at 9B tokens, with a 91.21% reduction in QK-path parameters and projection FLOPs. These results demonstrate that the shared routing trunk with private residual branches provides an effective additional compression layer on top of MLA-style compressed-KV attention: it preserves the benefits of MLA-style compressed-KV attention while further reducing the query-key content path and improving validation performance.

Inference performance. We additionally profile the inference behavior of the SmolLM-style MLA baseline and the QK-coupled variant on the 130M and 350M models. We measure prefill latency, recompute-based decode throughput, peak allocated memory, and the estimated latent KV-cache size. The results are summarized in Table 2. QK-coupled parameterization reduces the latent KV-cache estimate from 56 to 36 elements on the 130M model and from 72 to 48 elements on the 350M model. This corresponds to 35.71% and 33.33% latent-cache reductions, respectively. The same runs also show faster prefill, higher recompute-based decode throughput, and lower peak allocated memory, indicating that the shared routing trunk with private residual branches improves the inference-side efficiency of MLA-style compressed-KV attention.

C.4 DeepSeek-Style Implementation of MLA-style Compressed-KV Attention Pre-Training

We further evaluate QK-coupling in a pre-training regime based on a DeepSeek-style implementation of MLA-style compressed-KV attention (Liu et al., 2024a). This setting is close to the large-scale

Table 13: MLA-style compressed-KV attention pre-training validation perplexity on SmolLM backbones.

Model	Variant	PPL@0.6B	PPL@1.2B	PPL@1.8B	PPL@2.4B	PPL@3.0B
130M	MLA baseline	**42.04**	**35.59**	32.20	30.72	30.45
130M	QK-coupling	42.21	35.67	**32.01**	**30.51**	**30.23**
Model	**Variant**	**PPL@1.8B**	**PPL@3.6B**	**PPL@5.4B**	**PPL@7.2B**	**PPL@9.0B**
350M	MLA baseline	30.85	26.38	24.34	23.59	23.49
350M	QK-coupling	**30.70**	**26.21**	**24.29**	**23.45**	**23.36**

Table 14: Setup and QK-path cost summary for a DeepSeek-style implementation of MLA-style compressed-KV attention.

Model	L	d_{model}	FFN hidden	H	Experts	Top-k	Routed/Shared hidden	B	Reported Tokens (Billion)
3B	12	1280	7168	16	64	6	896/1792	512	200.00
9B	18	1920	10240	24	64	6	1280/2560	256	59.77
Model	**Variant**	r_Q^{lat} / r_s	d_c / d_s	d_{pos} or d'_{pos}	r_q	d_{kp}	**QK Params (M)**	**QK TFLOPs /iter**	**QK Cost Saving**
3B	MLA	1536	512	64	0	0	94.37	1187.47	0.00%
3B	QK-coupling (aggressive)	384	448	48	64	64	43.94	552.92	53.44%
3B	QK-coupling (conservative)	384	480	64	80	80	46.65	586.93	50.57%
9B	MLA	1536	512	64	0	0	212.34	1335.91	0.00%
9B	QK-coupling (conservative)	384	448	64	80	80	100.20	630.38	52.81%

MLA-style compressed-KV attention architecture used in our profiling study: the attention path already uses compressed query and key-value latent projections, and the feed-forward stack uses expert routing. We consider pre-training tasks on 3B and 9B MoE models. In the QK-coupled variants, the query-key content path is replaced by a shared routing trunk together with lightweight private residual branches implementing query-private / key-private correction. Following the score-space analysis in Theorem 3, we use joint query-private / key-private correction in the MLA-style compressed-KV attention regime because the key content path is constrained by the compressed latent representation.

C.4.1 3B AND 9B PRE-TRAINING COMPARISON

For the 3B experiments, the global batch size is $B = 512$ and the sequence length is $T = 4096$; therefore each training step corresponds to $512 \times 4096 = 2.10$M tokens. The reported 3B trajectory spans checkpoints from 10K to 95,368 steps, corresponding to 20.97B to 200.00B training tokens. For the 9B experiments, the global batch size is $B = 256$ and the sequence length is $T = 4096$; each step corresponds to $256 \times 4096 = 1.05$M tokens. The reported 9B trajectory spans checkpoints from 10K to 57K steps, corresponding to 10.49B to 59.77B training tokens. We report the validation trajectories against training tokens in Figures 6 and 8; for the 3B runs, the legend additionally reports the final test PPL.

Table 14 summarizes the model setup and the QK-path costs. The column r_Q^{lat}/r_s reports the baseline query-latent rank and the QK-coupled shared routing rank, respectively. The reported parameters and TFLOPs correspond to the MLA-style compressed-KV attention QK content-branch projection proxy. Since the projection FLOPs are proportional to the same QK parameter proxy under a fixed batch and sequence length, the parameter and projection-FLOP saving ratios are identical. On the 3B model, the aggressive and conservative QK-coupled variants reduce the QK-path projection-FLOP proxy by 53.44% and 50.57%, respectively. On the 9B model, the conservative QK-coupled variant reduces the same proxy by 52.81%.

Figures 6 and 8 show that QK-coupling provides a strong quality-efficiency trade-off in the setting based on a DeepSeek-style implementation of MLA-style compressed-KV attention. On the 3B model, the conservative QK-coupled configuration reaches the best final test PPL while reducing the QK-path projection-FLOP proxy by 50.57%. The aggressive configuration increases the QK-path saving to 53.44% and keeps the final test PPL within 0.5% of the MLA-style compressed-KV attention



Figure 8: Validation perplexity of the 9B pre-training runs based on a DeepSeek-style implementation of MLA-style compressed-KV attention. The horizontal axis reports training tokens.

Table 15: 3B query-private / key-private correction placement ablation.

Variant	r_s	d_s	d_{pos} **or** d'_{pos}	r_q	d_{kp}	**QK Params (M)**	**QK TFLOPs /iter**	**Final PPL**
balanced	384	448	48	64	64	43.94	552.92	**10.5827**
query-private	384	448	48	128	0	43.94	552.92	10.7193
key-private	384	448	48	0	128	43.94	552.92	10.6736
key-heavy	384	448	48	32	96	43.94	552.92	10.6002
query-heavy	384	448	48	96	32	43.94	552.92	10.6137

baseline. On the 9B model, QK-coupling achieves lower validation perplexity at every reported token checkpoint and reduces the QK-path cost by 52.81%.

C.4.2 3B Query-Private / Key-Private Correction Placement Ablation

To study how to place query-private / key-private correction in the MLA-style compressed-KV attention regime, we conduct a controlled ablation on the 3B model. The shared routing rank and shared cached latent dimension are fixed for the matched-cost variants, and we vary only the placement of private capacity between query-private correction and key-private correction. This isolates the effect of query-private correction, key-private correction, and joint query-private / key-private correction under the same QK-path parameter and projection-FLOP budget.

Table 15 compares different query-private / key-private correction placements under a matched private-rank budget. Among the matched-cost variants, the balanced joint query-private / key-private correction obtains the best final perplexity. The key-heavy and query-heavy variants also outperform the one-sided placements, supporting the use of joint query-private / key-private correction in MLA-style compressed-KV attention, where the key content path is already compressed by the latent representation.

C.5 LoRA-Style Fine-Tuning Experiments

We further evaluate QK-coupling in the LoRA-style fine-tuning regime. For QK-coupled adapter training, the pre-trained backbone is frozen and the low-rank adaptation parameters are trainable. The full-rank and GaLore controls update the full Q/K matrices, as specified below. We consider two representative fine-tuning settings: RoBERTa-Base on the GLUE benchmark and Qwen3.5-35B-A3B (Qwen Team, 2026) on a collection of math, commonsense, code, language modeling, and MMLU evaluation tasks. For parameter accounting, we report the theoretical number of non-frozen parameters on the QK path.

Table 16: Hyperparameters used in fine-tuning RoBERTa-Base on the GLUE benchmark.

	SST-2	CoLA	MRPC	STS-B	RTE	QNLI	QQP	MNLI
Batch size B	32	16	16	16	16	16	16	16
Learning Rate				1E-5				
Adapter Rank				16				
Shared Routing Rank				8				
Query-private Rank				4				

Table 17: Evaluation results of fine-tuning RoBERTa-Base on the GLUE benchmark. The QK parameter column reports theoretical non-frozen QK parameters only. The highest score in each group is in bold.

Algorithm	QK Params (M)	SST-2	CoLA	MRPC	STS-B	RTE	QNLI	QQP	MNLI	Avg
Full-Rank	14.16	94.61	56.54	92.63	90.06	77.98	92.33	91.42	86.01	85.20
ReLoRA	0.59	94.27	55.73	91.07	89.16	76.53	91.69	**91.05**	85.05	84.32
GaLore	14.16	93.34	57.66	90.71	87.76	75.81	91.51	89.59	**86.27**	84.08
QK-coupling (ours)	0.29	**94.38**	**58.08**	**91.73**	**89.37**	**79.06**	**91.82**	90.93	85.70	**85.13**

C.5.1 RoBERTa-Base on GLUE

For encoder-style language understanding, we fine-tune RoBERTa-Base (Liu et al., 2019) on the GLUE benchmark (Wang et al., 2018). We compare QK-coupling with full-rank fine-tuning, ReLoRA (Lialin et al., 2023), and GaLore (Zhao et al., 2024). The hyperparameters are summarized in Table 16. For QK parameter accounting, RoBERTa-Base has $L = 12$ layers and $d_{\text{model}} = d_{\text{qk}} = 768$. Full-rank fine-tuning and GaLore update full Q/K matrices, giving $2Ld_{\text{model}}d_{\text{qk}}$ trainable QK parameters. ReLoRA uses independent rank-r Q/K low-rank updates with $2Lr(d_{\text{model}} + d_{\text{qk}})$ parameters. QK-coupling uses shared routing rank r_s and query-private rank r_q, giving $L(d_{\text{model}}r_s + 2r_sd_{\text{qk}} + d_{\text{model}}r_q + r_qd_{\text{qk}})$ trainable QK parameters, which reduces to $Ld_{\text{model}}(3r_s + 2r_q)$ in the square case.

Table 17 shows that QK-coupling achieves the best average score among parameter-efficient baselines on GLUE. Compared with ReLoRA, QK-coupling uses half of the theoretical trainable QK parameters (0.29M vs. 0.59M) and improves the average score from 84.32 to 85.13. Compared with GaLore, QK-coupling trains far fewer QK-path parameters while also improving the average score. Its average performance is close to full-rank fine-tuning, indicating that the QK-coupled update is an effective adaptation structure for encoder-style language understanding tasks.

C.5.2 Qwen3.5-35B-A3B Fine-Tuning

For large decoder-style fine-tuning, we evaluate QK-coupling on Qwen3.5-35B-A3B. The evaluation covers GSM8K (Cobbe et al., 2021), BoolQ (Clark et al., 2019), WikiText (Merity et al., 2016), HumanEval (Chen et al., 2021), and MMLU (Hendrycks et al., 2020) categories. We compare QK-coupling with ReLoRA, LoRA (Hu et al., 2022), and ThinKeys (Yao et al., 2026). The hyperparameters are summarized in Table 18. The main evaluation results are provided in Table 3, and the full MMLU breakdown is provided in Table 19.

Table 3 summarizes the main Qwen3.5-35B-A3B fine-tuning results. QK-coupling achieves the best or tied-best score on all reported task groups while using fewer full-attention QK adapter parameters than the baseline low-rank QK updates. In particular, QK-coupling matches or improves the reported aggregate results on GSM8K, BoolQ, WikiText, HumanEval, and MMLU in Table 3. These results indicate that the QK-coupled update remains effective for large decoder-style fine-tuning.

Table 19 provides the full Qwen3.5-35B-A3B evaluation across all MMLU subcategories and auxiliary tasks. The detailed breakdown shows competitive QK-coupling performance across multiple categories: QK-coupling remains competitive across humanities, STEM, social sciences, and other MMLU subsets, while also improving or matching the aggregate results shown in Table 3.

Table 18: Hyperparameters for Qwen3.5-35B-A3B fine-tuning.

Dataset	GSM8K	BoolQ	MMLU	WikiText	HumanEval
Global batch size B	64	64	64	64	64
Learning Rate	2E-5	2E-5	2E-5	5E-6	2E-5
Adapter Rank			2		
Shared Routing Rank			1		
Query-private Rank			1		

Table 19: Full evaluation of Qwen3.5-35B-A3B. For WikiText, smaller is better. For other tasks, larger is better.

Dataset	QK-coupling	ReLoRA	LoRA	ThinKeys	Dataset	QK-coupling	ReLoRA	LoRA	ThinKeys
GSM8K(exact_match)	**87.57**	83.49	87.49	87.34	BoolQ	**91.47**	88.56	91.31	90.46
WikiText(word_perplexity)(↓)	**8.303**	8.304	12.69	13.04	HumanEval(pass@1)	**40.24**	37.80	33.54	36.59
Abstract Algebra	64.00	**71.00**	59.00	58.00	Anatomy	82.96	82.22	**84.44**	81.48
Astronomy	89.47	90.13	**90.79**	90.13	Business Ethics	**88.00**	85.00	85.00	87.00
Clinical Knowledge	90.57	**91.32**	**91.32**	90.57	College Biology	**94.44**	93.75	93.75	**94.44**
College Chemistry	**67.00**	65.00	59.00	58.00	College Computer Science	**74.00**	70.00	70.00	68.00
College Mathematics	73.00	**75.00**	72.00	72.00	College Medicine	**84.97**	84.39	81.50	80.92
College Physics	**72.55**	**72.55**	68.63	68.63	Computer Security	86.00	85.00	**88.00**	**88.00**
Conceptual Physics	**94.04**	93.19	93.19	92.77	Econometrics	81.58	81.58	**82.46**	79.82
Elec. Engineering	83.45	84.14	**84.83**	84.14	Elementary Mathematics	**78.84**	78.57	78.57	76.72
Formal Logic	67.46	**69.84**	66.67	67.46	Global Facts	54.00	**55.00**	53.00	52.00
HS Biology	94.52	95.16	**95.48**	95.16	HS Chemistry	81.77	**82.27**	81.77	80.30
HS Computer Science	**94.00**	93.00	**94.00**	**94.00**	HS European Hist.	86.67	86.67	**87.88**	87.27
HS Geography	93.43	**93.94**	92.42	92.93	HS Gov/Politics	97.93	97.41	97.93	**98.45**
HS Macroeconomics	**90.77**	89.74	88.21	87.69	HS Mathematics	61.11	**61.48**	**61.48**	58.89
HS Microeconomics	**96.64**	96.22	96.22	95.80	HS Physics	78.15	**78.81**	75.50	76.82
HS Psychology	95.41	95.41	**95.78**	95.60	HS Statistics	81.94	81.02	**83.33**	82.41
HS US History	94.61	94.61	**95.10**	**95.10**	HS World History	93.25	**94.09**	**94.09**	92.83
Human Aging	80.72	80.72	**81.61**	80.72	Human Sexuality	**90.08**	89.31	**90.08**	**90.08**
International Law	**90.91**	**90.91**	90.08	89.26	Jurisprudence	90.74	**91.67**	90.74	90.74
Logical Fallacies	87.73	**88.34**	85.89	85.89	Machine Learning	**77.68**	76.79	75.00	75.00
Management	90.29	89.32	**92.23**	91.26	Marketing	**93.16**	**93.16**	**93.16**	**93.16**
Medical Genetics	**94.00**	**94.00**	**94.00**	**94.00**	Miscellaneous	**93.49**	93.10	93.23	93.36
MMLU	**83.39**	82.99	82.88	81.86	Moral Disputes	**84.10**	83.24	**84.10**	83.53
Moral Scenarios	**68.60**	63.13	65.36	53.74	Nutrition	**88.89**	88.24	**88.89**	**88.89**
Philosophy	86.82	86.82	**88.10**	87.14	Prehistory	**90.74**	90.12	90.43	**90.74**
Prof. Accounting	75.53	75.18	76.24	**76.95**	Prof. Law	68.12	68.32	68.25	**68.45**
Prof. Medicine	93.38	**94.12**	93.38	92.65	Prof. Psychology	89.22	**89.54**	88.73	88.40
Public Relations	74.55	**75.45**	73.64	74.55	Security Studies	**84.90**	84.49	83.67	84.08
Sociology	**92.54**	92.04	91.54	92.04	US Foreign Policy	90.00	90.00	**91.00**	**91.00**
Virology	**58.43**	57.83	55.42	56.63	World Religions	91.81	**92.40**	90.64	90.64

C.6 Additional Training Results

This subsection collects matched-run results, additional QK controls and rank allocations, and end-to-end training measurements. For the 130M matched runs and the 3B baseline/conservative comparison, $\pm$ denotes sample standard deviation across runs of the same configuration, using denominator $n-1$. The aggressive 3B aggregate is reported in the same numerical format as Figure 6. Paired improvements are baseline PPL minus QK-coupled PPL for the corresponding runs.

C.6.1 Multi-Seed Results

Baseline provenance. The 60M–1.3B full-rank results underlying Figure 5 and Table 9 were obtained from our same-codebase reruns. The full-rank and QK-coupled models at each scale were trained on the same $8\times$A100 machine, following the same per-scale training and evaluation protocol apart from the QK parameterization.

Final results at 130M and 3B. Table 20 reports the individual matched runs and their summaries, followed by the aggregate aggressive result. At 130M, all three QK-coupled runs improve over the corresponding full-rank runs, with a mean paired improvement of 0.4201 ± 0.2029 PPL. The first pair is the result used in the original scaling comparison; the additional runs retain the same training and evaluation protocol. At 3B, all three matched full-length runs, with seeds 1234/1235/1236, favor conservative QK-coupling. Its final test PPL is 10.5148 ± 0.0078, compared with 10.5549 ± 0.0095 for the MLA baseline, and the paired improvement is 0.0401 ± 0.0018 PPL. The aggressive configuration has the reported three-run final-test summary 10.5895 ± 0.0107 ($n = 3$). Relative to the baseline mean, the aggressive mean is higher by 0.0346 PPL (0.33%), while the QK-path saving increases from 50.57% for conservative compression to 53.44% for aggressive compression. Figure 6 retains

Table 20: Multi-run PPL results. The 130M values are validation PPL; the 3B values are final test PPL after 200B tokens. The matched baseline/conservative summaries report mean ± sample SD. The separate aggressive aggregate follows the final-test summary in Figure 6. Positive paired improvement favors QK-coupling. The 130M runs are indexed in their reported order.

Model	run	Baseline	QK-coupling	Paired improvement
130M	Run 1	23.9188	23.2669	0.6519
	Run 2	23.8928	23.5592	0.3336
	Run 3	23.9011	23.6262	0.2749
	Mean ± SD	23.9042 ± 0.0133	23.4841 ± 0.1911	0.4201 ± 0.2029
3B	Run 1	10.5454	10.5074	0.0380
	Run 2	10.5549	10.5140	0.0409
	Run 3	10.5643	10.5229	0.0414
	Mean ± SD	10.5549 ± 0.0095	10.5148 ± 0.0078	0.0401 ± 0.0018
3B QK-coupling aggressive		10.5895 ± 0.0107 ($n = 3$; reported aggregate)		

Table 21: Converged Independent-LR control under the 350M protocol of Table 11. Both methods use the same 50% QK parameter and projection-FLOP budget. Only Independent-LR's learning rate is reduced; the QK-coupling row repeats the original comparison for reference.

Method	Learning rate	PPL@10K	PPL@20K	PPL@40K
Independent-LR (tuned)	1.0×10^{-3}	23.93	19.55	18.19
QK-coupling	1.5×10^{-3}	22.55	19.34	17.64

the original seed-1234 validation trajectories and reports the aggregate final-test statistics in its legend. The original aggressive value 10.5827 remains the individual-run entry in the matched-placement comparisons in Tables 4 and 15.

Aligned intermediate checkpoint. At the aligned 20K-step checkpoint, three matched 3B runs give PPL 13.3423 ± 0.0479 for the MLA baseline and 13.3290 ± 0.0223 for conservative QK-coupling. These intermediate-checkpoint results are reported separately from the final-test comparison above.

C.6.2 Converged Independent-LR Control

We use the matched-budget protocol in Appendix C.2.4 and add the converged Independent-LR comparison in Table 21. Reducing only its learning rate from 1.5×10^{-3} to 1.0×10^{-3} yields PPL $23.93/19.55/18.19$ at 10K/20K/40K steps. QK-coupling retains its original learning rate of 1.5×10^{-3} and obtains $22.55/19.34/17.64$ at the same checkpoints. Thus, the converged Independent-LR control remains worse than QK-coupling under the same QK parameter and projection-FLOP budget.

Under the original common-learning-rate protocol, Independent-LR obtained PPL $681.39/692.47/675.95$ at 10K/20K/40K steps; the main comparison table reports its tuned, converged results. A short 3K-step sweep at learning rates 3×10^{-4}, 6×10^{-4}, 1.0×10^{-3}, and 1.5×10^{-3} gives PPL $44.03/36.59/34.38/33.59$, respectively. Two additional full-length runs at 1.5×10^{-3} end at PPL 621.16 and 593.89. These results distinguish the favorable early descent at the larger learning rate from its long-horizon sensitivity, and motivate reporting the tuned, converged comparison alongside the original protocol.

C.6.3 Shared-Only Controls and Rank Allocation

350M matched-budget allocation. Table 22 varies the shared/query-private ranks under approximately the same 50% QK parameter and projection-FLOP budget as Table 11. All configurations retain the original 1.5×10^{-3} learning rate and 40K-step training horizon. The shared-only configuration obtains PPL 17.7677, compared with 17.6400 for the default shared–private configuration and 18.0500 for full-rank QK. All four allocations improve over full-rank QK, while the default provides the best observed balance of shared and private capacity.

Table 22: 350M shared/query-private allocation at approximately 50% QK budget. All rows use $r_k = 0$. Full-rank QK obtains PPL 18.0500 under this protocol.

Allocation	r_s	r_q	Final validation PPL
Default	256	128	**17.6400**
Shared-heavy / private-light	304	56	17.7048
Shared-only	341	0	17.7677
Shared-light / private-heavy	208	200	17.8854

Table 23: A more aggressive 130M rank configuration. QK-side savings refer to the matrix-parameter and projection-FLOP counts. These are individual-run results under the original 130M protocol.

Configuration	r_s	r_q	QK-side saving	Validation PPL
Full-rank	–	–	0.0%	23.9188
Default	96	48	75.0%	**23.2669**
Aggressive	64	32	83.3%	23.6950

3B shared-only control. For seed 1236 at the aligned 20K-step checkpoint, the shared-only control obtains PPL 13.5874, while the MLA baseline and conservative shared–private model obtain 13.3525 and 13.3263, respectively. This comparison complements the 350M allocation study and the full-length placement ablation in Table 15.

130M compression trade-off. Table 23 compares the original width-relative ranks with a more aggressive setting under the same 130M protocol. Reducing (r_s, r_q) from $(96, 48)$ to $(64, 32)$ increases QK-side savings from 75.0% to 83.3%, while PPL changes from 23.2669 to 23.6950. Both rank configurations improve over the corresponding full-rank PPL of 23.9188, providing two observed quality–compression operating points.

C.6.4 End-to-End Training Efficiency

1.3B standard-attention training. We measure training on 8×A100-80GB GPUs with data parallelism. QK-coupling reaches the full-rank model's 100K-step validation PPL of 15.56 at 73.5K steps. The measured mean step times are 1.981 s and 1.949 s for full-rank and QK-coupled training. Estimating time-to-quality as steps-to-target times mean step time gives 55.03 h and 39.79 h, a 27.7% reduction. Peak memory measured by `nvidia-smi` decreases from 39.66 to 37.40 GiB, or 5.7%; see Table 24.

3B MLA/MoE training. On the 64-GPU run, with `PP=TP=EP=1` and `EP_INNER=8`, conservative QK-coupling increases global throughput from 50.42K to 52.88K tokens/s, a 4.89% improvement. Per-GPU peak memory decreases from 73,096.14 to 69,592.38 MiB, a 4.79% reduction. Table 25 reports these measured training quantities separately from the analytical QK-path costs in Table 14. These measurements show that QK-path compression also translates into improved throughput and lower training memory on the MLA/MoE stack.

D Theoretical Results and Proofs

This section gives formal statements and proofs corresponding to the three theoretical results in the main text. Appendix D.1 corresponds to Theorem 1 and connects multi-anchor QK-coupling to a shared low-dimensional relational representation. Appendix D.2 corresponds to Theorem 2 and characterizes how common-core-preserving MLA-style compression can increase QK-coupling. Appendix D.3 corresponds to Theorem 3 and characterizes the complementary score-space correction directions provided by query-private and key-private residual branches.

We write $\|\cdot\|_F$ and $\|\cdot\|_2$ for the Frobenius and spectral norms, respectively, and $\dagger$ for the Moore–Penrose pseudoinverse. Singular values and eigenvalues are ordered non-increasingly. The approximate rank of the relational space is denoted by p, while r_s, r_q, r_k denote the shared, query-private, and key-private ranks used by the parameterization.

Table 24: End-to-end 1.3B training measurements. The common validation-PPL target is 15.56. Time-to-quality is estimated from steps-to-target and measured mean step time.

Method	Step time (s)	Steps to target	Estimated time to target (h)	Peak memory (GiB)
Full-rank	1.981	100K	55.03	39.66
QK-coupling	1.949	73.5K	39.79	37.40

Table 25: Measured 3B training throughput and memory on the 64-GPU run. Throughput is global; peak memory is per GPU. These measurements correspond to the conservative QK-coupled configuration.

Method	Global throughput (K tokens/s)	Peak memory (MiB/GPU)
MLA baseline	50.42	73,096.14
QK-coupling conservative	52.88	69,592.38
Relative change	$+4.89\%$	-4.79%

D.1 Multi-Anchor Coupling and Shared Low-Dimensional Relational Representations

D.1.1 Notation and Analysis Object

For an operator $A \in \{Q, K\}$, let $A_1, \dots, A_L$ denote the nonzero activations after the same preprocessing as in Appendix C.1. Activations from the same operator have a consistent shape across layers, while Q and K may retain their native output widths.

Define the layer-relational matrix by $(G_A)_{\ell a} = \langle A_\ell, A_a \rangle_F / (\|A_\ell\|_F \|A_a\|_F)$ for $1 \le \ell, a \le L$. For anchor a, remove the self-similarity entry and define the relational trajectory $u_A^{(a)} = ((G_A)_{\ell a})_{\ell \neq a} \in \mathbb{R}^{L-1}$. We consider nonzero Q/K anchor trajectories and use the main-text metric $\kappa_{QK}(a) = \langle u_Q^{(a)}, u_K^{(a)} \rangle / (\|u_Q^{(a)}\|_2 \|u_K^{(a)}\|_2)$, with $\bar{\kappa} := \bar{\kappa}_{QK} = L^{-1} \sum_{a=1}^{L} \kappa_{QK}(a)$.

Let $D_A = \mathrm{diag}(\|u_A^{(1)}\|_2, \dots, \|u_A^{(L)}\|_2)$ and $C_A = (G_A - I_L) D_A^{-1}$. Thus, the a-th column of C_A is the unit-normalized anchor trajectory with a zero inserted at the anchor position. Define the common and differential relational components by $M = (C_Q + C_K)/2$ and $D = (C_Q - C_K)/2$.

For $1 \le p \le L$, let $U_p \in \mathbb{R}^{L \times p}$ contain the first p left singular vectors of M, and set $P_p = U_p U_p^\top$ and $\tau_p = L^{-1} \sum_{j>p} \sigma_j(M)^2$. The quantity τ_p is the normalized truncation error of the common relational component.

D.1.2 Theorem and Proof

Theorem 4 (Coupling and shared low-dimensional relational representation)**.** *Under the definitions above, the common and differential components satisfy*

$$\frac{\|M\|_F^2}{L} = \frac{1 + \bar{\kappa}}{2}, \qquad \frac{\|D\|_F^2}{L} = \frac{1 - \bar{\kappa}}{2}. \tag{19}$$

Let $B_A = U_p^\top C_A$. The reconstructions using the same left factor, $\widehat{C}_Q = U_p B_Q$ and $\widehat{C}_K = U_p B_K$, have normalized joint error

$$\mathcal{R}_p := \frac{\|C_Q - \widehat{C}_Q\|_F^2 + \|C_K - \widehat{C}_K\|_F^2}{2L} = \tau_p + \frac{\|(I_L - P_p) D\|_F^2}{L} \le \tau_p + \frac{1 - \bar{\kappa}}{2} \le 1. \tag{20}$$

Define the optimal shared error

$$\mathcal{E}_p^{\mathrm{sh}} = \frac{1}{2L} \min_{\substack{U^\top U = I_p \\ B_Q, B_K}} \left(\|C_Q - U B_Q\|_F^2 + \|C_K - U B_K\|_F^2 \right), \tag{21}$$

$$\mathcal{E}_p^{\mathrm{ind}} = \frac{1}{2L}\left(\min_{\substack{U_Q^\top U_Q = I_p \\ B_Q}} \|C_Q - U_Q B_Q\|_F^2 + \min_{\substack{U_K^\top U_K = I_p \\ B_K}} \|C_K - U_K B_K\|_F^2\right). \tag{22}$$

Let $H_Q = C_Q C_Q^\top$, $H_K = C_K C_K^\top$, *and* $\Phi_p(H) = \sum_{j=1}^p \lambda_j(H)$. *Then*

$$\mathcal{E}_p^{\mathrm{sh}} = 1 - \frac{\Phi_p(H_Q + H_K)}{2L} \le \mathcal{R}_p, \tag{23}$$

and the additional error incurred by sharing relative to independent representations is

$$\mathcal{E}_p^{\mathrm{sh}} - \mathcal{E}_p^{\mathrm{ind}} = \frac{\Phi_p(H_Q) + \Phi_p(H_K) - \Phi_p(H_Q + H_K)}{2L} \ge 0. \tag{24}$$

Proof of Theorem 4. Because every column is normalized, $\|C_Q\|_F^2 = \|C_K\|_F^2 = L$ and $\langle C_Q, C_K\rangle_F = L\bar{\kappa}$. Expanding the squared norms gives $\|M\|_F^2 = \frac{1}{4}\left(\|C_Q\|_F^2 + \|C_K\|_F^2 + 2\langle C_Q, C_K\rangle_F\right) = \frac{L}{2}(1+\bar{\kappa})$, and similarly $\|D\|_F^2 = L(1-\bar{\kappa})/2$, which proves Eq. (19).

Since $\widehat{C}_A = P_p C_A$, using $C_Q = M + D$ and $C_K = M - D$ yields cancellation of the cross terms:

$$\|(I_L - P_p)C_Q\|_F^2 + \|(I_L - P_p)C_K\|_F^2 = 2\|(I_L - P_p)M\|_F^2 + 2\|(I_L - P_p)D\|_F^2.$$

By the definition of P_p, $\|(I_L - P_p)M\|_F^2 = \sum_{j>p}\sigma_j(M)^2 = L\tau_p$. Since an orthogonal projection is contractive, $\|(I_L - P_p)D\|_F^2 \le \|D\|_F^2$. Hence $\mathcal{R}_p \le \tau_p + (1-\bar{\kappa})/2$. Moreover, $\tau_p \le \|M\|_F^2/L = (1+\bar{\kappa})/2$, giving the final inequality in Eq. (20).

We next derive the exact optimal shared error. For fixed column-orthonormal U, the decomposition

$$\|C_A - UB_A\|_F^2 = \|(I_L - UU^\top)C_A\|_F^2 + \|U^\top C_A - B_A\|_F^2$$

shows that the optimal coefficients are $B_A = U^\top C_A$. Consequently,

$$2L\mathcal{E}_p^{\mathrm{sh}} = 2L - \max_{U^\top U = I_p} \mathrm{tr}\big(U^\top (H_Q + H_K)U\big).$$

For a positive semidefinite matrix $H = \sum_j \lambda_j v_j v_j^\top$, let $w_j = \|U^\top v_j\|_2^2$. Then $\mathrm{tr}(U^\top H U) = \sum_j \lambda_j w_j$, with $0 \le w_j \le 1$ and $\sum_j w_j = p$. Ordering the eigenvalues non-increasingly gives an upper bound $\sum_{j=1}^p \lambda_j$, attained by $U = [v_1, \ldots, v_p]$. Thus the maximum trace equals $\Phi_p(H)$, which proves Eq. (23). This is the Ky Fan variational principle applied to the joint relational covariance.

When Q and K choose independent subspaces, the same argument gives $\mathcal{E}_p^{\mathrm{ind}} = 1 - [\Phi_p(H_Q) + \Phi_p(H_K)]/(2L)$. Subtracting this expression from Eq. (23) gives Eq. (24). Finally,

$$\Phi_p(H_Q + H_K) = \max_{U^\top U = I_p} \left[\mathrm{tr}(U^\top H_Q U) + \mathrm{tr}(U^\top H_K U)\right] \le \Phi_p(H_Q) + \Phi_p(H_K),$$

so the sharing gap is nonnegative. □

Connection to the main-text Theorem 1 and the 0.338 observation. Theorem 4 gives a quantitative decomposition of multi-anchor QK-coupling into common and differential relational components. For Qwen2.5-3B Base reported in the main text, $\bar{\kappa} = 0.338$ gives $\|M\|_F^2/L = 0.669$ and $\|D\|_F^2/L = 0.331$ by Eq. (19). These values quantify common and differential energy in unit-normalized anchor-relative trajectories; parameter, storage, and projection-FLOP savings are accounted for separately in Appendix F. The corresponding mean squared distance between the unit-normalized Q/K trajectories is $2(1-\bar{\kappa}) = 1.324$, which is 33.8% below the orthogonal reference of 2. For a shared p-dimensional relational subspace, the theorem further gives $\mathcal{R}_p \le \tau_p + 0.331$: the quality of shared approximation is determined jointly by the common component's spectral tail and the relational discrepancy not captured by the shared subspace. This provides a quantitative motivation for representing common relational components while retaining operator-specific differences.

Theorem 4 quantifies sharing in the layer-relational space. Proposition 1 complements this analysis at the channel level by characterizing the input-weighted approximation error of a shared factor with independent Q/K readouts. The selected channel ranks are evaluated in the matched-budget experiments.

D.1.3 Shared Channel Projection and Joint Approximation

Proposition 1 (Input-weighted optimal shared low-rank projection)**.** *Let $x \in \mathbb{R}^{1\times d}$ be a random input with finite second moment, and let $\Sigma = \mathbb{E}[x^\top x] \succeq 0$. Given $W_Q \in \mathbb{R}^{d\times d_Q}$ and $W_K \in \mathbb{R}^{d\times d_K}$, define $F = \Sigma^{1/2}[W_Q\ W_K]$. For any shared channel rank $r_s \geq 0$,*

$$\min_{A^S, B_Q, B_K} \mathbb{E}\left[\|xW_Q - xA^SB_Q\|_2^2 + \|xW_K - xA^SB_K\|_2^2\right] = \sum_{j>r_s} \sigma_j(F)^2, \tag{25}$$

where $A^S \in \mathbb{R}^{d\times r_s}$ and both readouts have r_s rows.

Proof of Proposition 1. Let $B = [B_Q\ \ B_K]$. The objective equals $\|F - \Sigma^{1/2}A^SB\|_F^2$. Since $\Sigma^{1/2}A^SB$ has rank at most r_s, the objective is lower bounded by the optimal rank-r_s approximation error of F.

For a direct verification, let R be any matrix of rank at most r_s, and let P_R project onto its column space. Then

$$\|F - R\|_F^2 \geq \|(I - P_R)F\|_F^2 = \|F\|_F^2 - \mathrm{tr}(P_RFF^\top) \geq \sum_{j>r_s} \sigma_j(F)^2.$$

The lower bound is attained by the truncated singular value decomposition F_{r_s}. Since $\mathrm{col}(F_{r_s}) \subseteq \mathrm{col}(F) \subseteq \mathrm{col}(\Sigma^{1/2})$, write $F_{r_s} = UV$ with $\mathrm{col}(U) \subseteq \mathrm{col}(\Sigma^{1/2})$. Taking $A^S = (\Sigma^{1/2})^\dagger U$ and $B = V$, with zero padding when needed, gives $\Sigma^{1/2}A^SB = F_{r_s}$. Therefore the lower bound is attained by a shared input factor with independent Q/K readouts. □

Theorem 4 characterizes shared approximation in the layer-relational space. At the channel level, this proposition provides a complementary input-weighted characterization of the proposed shared-factor parameterization: its optimal approximation error is determined by the joint channel spectrum. Together, these results motivate the shared-routing design, whose quality at the selected channel ranks is assessed by the matched-budget experiments.

D.2 Common-Core-Preserving MLA Compression and Coupling Gain

D.2.1 Common-Core Decomposition Model

This subsection corresponds to Theorem 2. We analyze an MLA-style compression mechanism through optimal low-rank KV reconstruction on fixed reference activations. This fixed-activation model isolates a sufficient mechanism for coupling gain through preservation of a common relational core. For layer ℓ, write the reference representations as $Q_\ell^{\mathrm{ref}} = Q_\ell^0 + N_\ell^Q$ and $K_\ell^{\mathrm{ref}} = K_\ell^0 + N_\ell^K$, where Q_ℓ^0 and K_ℓ^0 denote the common relational core and N_ℓ^Q, N_ℓ^K denote operator-specific components. The matrices K_ℓ and V_ℓ may include concatenated content outputs from all heads; this concatenation represents the joint latent used by the compressed KV path.

Condition D.2-A (Common relational and activation separation)**.** All core activations are nonzero, and there exists a common relational matrix G_0 such that $\cos_F(Q_\ell^0, Q_a^0) = \cos_F(K_\ell^0, K_a^0) = (G_0)_{\ell a}$. For $A \in \{Q, K\}$, assume core-residual cross-layer orthogonality $\langle A_\ell^0, N_a^A\rangle_F = 0$ for all ℓ, a. Further, assume that there is a layer-independent $\rho_A \geq 0$ such that $\|N_\ell^A\|_F^2 = \rho_A\|A_\ell^0\|_F^2$. Define the residual relational matrix by $(R_A)_{\ell a} = \langle N_\ell^A, N_a^A\rangle_F/(\|A_\ell^0\|_F\|A_a^0\|_F)$.

Condition D.2-B (Spectral dominance of the joint KV core)**.** Let $Y_\ell^{\mathrm{ref}} = [K_\ell^{\mathrm{ref}}\ V_\ell^{\mathrm{ref}}] = Y_\ell^0 + N_\ell^Y = X_\ell W_\ell$, where $Y_\ell^0 = [K_\ell^0\ V_\ell^0]$. Assume $\mathrm{rank}(Y_\ell^0) = d_c$, $(Y_\ell^0)^\top N_\ell^Y = 0$, $Y_\ell^0(N_\ell^Y)^\top = 0$, and $\sigma_{d_c}(Y_\ell^0) > \|N_\ell^Y\|_2$. Define the optimal rank-constrained representation by $\widehat{Y}_\ell \in \arg\min_{\mathrm{rank}(Y)\leq d_c} \|Y_\ell^{\mathrm{ref}} - Y\|_F^2$. The compressed query remains Q_ℓ^{ref}, while the key uses the key part of $\widehat{Y}_\ell$.

Condition D.2-C (Relational component separation)**.** For anchor a, let $g_a = (G_0)_{-a,a}$, $e_{Q,a} = (R_Q)_{-a,a}$, and $e_{K,a} = (R_K)_{-a,a}$. Assume $g_a \neq 0$ and $\langle g_a, e_{Q,a}\rangle = \langle g_a, e_{K,a}\rangle = 0$. The two residuals may be correlated. Define $\nu_Q(a) = \|e_{Q,a}\|_2^2/\|g_a\|_2^2$, $\nu_K(a) = \|e_{K,a}\|_2^2/\|g_a\|_2^2$, and $\beta_a = \langle e_{Q,a}, e_{K,a}\rangle/\|g_a\|_2^2$.

D.2.2 THEOREM AND PROOF

Theorem 5 (Coupling gain from common-core-preserving compression)**.** *Under Conditions D.2-A–D.2-C, the optimal rank-constrained representation exactly preserves the common core, namely* $\widehat{Y}_\ell = Y^0_\ell$*. Moreover, there exists an input-generated latent with* d_c *columns,* $C^{KV}_\ell = X_\ell W^{DKV}_\ell$*, such that* $Y^0_\ell = C^{KV}_\ell [B_{K,\ell}\ B_{V,\ell}]$*.*

The pre- and post-compression coupling scores satisfy

$$\kappa_{\mathrm{ref}}(a) = \frac{1+\beta_a}{\sqrt{(1+\nu_Q(a))(1+\nu_K(a))}}, \qquad \kappa_{\mathrm{cmp}}(a) = \frac{1}{\sqrt{1+\nu_Q(a)}}. \tag{26}$$

Their difference is

$$\Gamma_a := \kappa_{\mathrm{cmp}}(a) - \kappa_{\mathrm{ref}}(a) = \frac{\sqrt{1+\nu_K(a)} - 1 - \beta_a}{\sqrt{(1+\nu_Q(a))(1+\nu_K(a))}}. \tag{27}$$

Consequently,

$$\kappa_{\mathrm{cmp}}(a) \geq \kappa_{\mathrm{ref}}(a) \quad \Longleftrightarrow \quad \beta_a \leq \sqrt{1+\nu_K(a)} - 1. \tag{28}$$

When the two operator-specific relational residuals are orthogonal, so that $\beta_a = 0$*,*

$$\kappa_{\mathrm{cmp}}(a) = \kappa_{\mathrm{ref}}(a)\sqrt{1+\nu_K(a)}. \tag{29}$$

If $e_{K,a} \neq 0$*, the gain at that anchor is strictly positive. If* $\beta_a = 0$ *for every anchor and at least one anchor has a nonzero key-specific relational residual, the mean coupling score strictly increases.*

Proof of Theorem 5. Fix a layer and omit the layer subscript. Take compact singular value decompositions of the core and residual: $Y^0 = U_0\Lambda_0 V_0^\top, \qquad N^Y = U_N\Lambda_N V_N^\top$. The two-sided orthogonality in Condition D.2-B makes their column and row spaces orthogonal, so

$$Y^{\mathrm{ref}} = [U_0\ U_N] \begin{bmatrix} \Lambda_0 & 0 \\ 0 & \Lambda_N \end{bmatrix} [V_0\ V_N]^\top.$$

Thus the nonzero singular values of Y^{ref} are the union of the two singular-value sets. Since the core has rank d_c and $\sigma_{d_c}(Y^0) > \|N^Y\|_2$, the largest d_c singular components all come from the core. The optimal rank-d_c approximation is therefore $\widehat{Y} = Y^0$.

To construct the input-generated latent, note that $N^Y V_0 = 0$, and hence $Y^{\mathrm{ref}} V_0 = U_0\Lambda_0$. Using $Y^{\mathrm{ref}} = XW$, take $C^{KV} = XWV_0$ and $[B_K\ B_V] = V_0^\top$. Then $C^{KV}[B_K\ B_V] = U_0\Lambda_0 V_0^\top = Y^0$, which proves the latent factorization after restoring the layer subscript.

We next express the activation decomposition in terms of relational trajectories. By Condition D.2-A, $\langle A^{\mathrm{ref}}_\ell, A^{\mathrm{ref}}_a\rangle_F = \langle A^0_\ell, A^0_a\rangle_F + \langle N^A_\ell, N^A_a\rangle_F$, and $\|A^{\mathrm{ref}}_\ell\|_F^2 = (1+\rho_A)\|A^0_\ell\|_F^2$. Hence $G^{\mathrm{ref}}_A = (G_0 + R_A)/(1+\rho_A)$ and $u^{(a),\mathrm{ref}}_A = (g_a + e_{A,a})/(1+\rho_A)$.

After compression, the key relational trajectory is g_a. Cosine similarity is invariant to positive scaling, so $\kappa_{\mathrm{ref}}(a) = \cos(g_a + e_{Q,a}, g_a + e_{K,a})$ and $\kappa_{\mathrm{cmp}}(a) = \cos(g_a + e_{Q,a}, g_a)$. By Condition D.2-C,

$$\langle g_a + e_{Q,a}, g_a + e_{K,a}\rangle = \|g_a\|_2^2(1+\beta_a),$$

while $\|g_a + e_{A,a}\|_2^2 = \|g_a\|_2^2(1+\nu_A(a))$ and $\langle g_a + e_{Q,a}, g_a\rangle = \|g_a\|_2^2$. Substitution into the cosine definition gives Eq. (26). Subtracting the two expressions gives Eq. (27), and the positivity of the denominator gives Eq. (28).

When $\beta_a = 0$, Eq. (26) gives Eq. (29), and

$$\Gamma_a = \frac{1}{\sqrt{1+\nu_Q(a)}}\left(1 - \frac{1}{\sqrt{1+\nu_K(a)}}\right).$$

This quantity is strictly positive when $\nu_K(a) > 0$. Averaging over anchors completes the proof. □

Connection to the main-text Theorem 2. Theorem 5 gives a direct coupling-level explanation for the MLA observation in the main text. The common KV latent preserves the dominant relational core, while the relative contribution of key-specific relational residuals is reduced by the compressed reconstruction. The compressed key trajectory retains the common relational structure, and coupling at anchor a is nondecreasing when $\beta_a \le \sqrt{1+\nu_K(a)} - 1$ also holds. In particular, coupling strictly increases when the two operator-specific relational residuals are orthogonal and the removed key-specific relational residual is nonzero.

This result gives the common content latent in the MLA instantiation a precise role: it provides a shared carrier for the routing structure that is common to Q and K. Separate Q/K readouts and private residual branches retain the remaining operator-specific degrees of freedom. The theorem therefore explains the empirical combination of stronger coupling and efficient shared routing through one common-core mechanism.

D.2.3 Stability under Relational Perturbations

Corollary 1 (Coupling gain under approximate core preservation)**.** *Keep the query trajectory fixed. Suppose the approximately compressed key trajectory can be written as* $\widetilde{u}_K^{(a)} = s_a(g_a + \delta_a)$ *with* $s_a > 0$ *and* $\|\delta_a\|_2 \le \epsilon_a \|g_a\|_2$ *for* $0 \le \epsilon_a < 1$*. Then*

$$|\widetilde{\kappa}(a) - \kappa_{\mathrm{cmp}}(a)| \le 2\epsilon_a, \tag{30}$$

and

$$\widetilde{\bar{\kappa}} - \bar{\kappa}_{\mathrm{ref}} \ge \frac{1}{L}\sum_a \Gamma_a - \frac{2}{L}\sum_a \epsilon_a. \tag{31}$$

Proof of Corollary 1. For $v = g + \delta \neq 0$, $\frac{v}{\|v\|_2} - \frac{g}{\|g\|_2} = \frac{v-g}{\|g\|_2} + v\left(\frac{1}{\|v\|_2} - \frac{1}{\|g\|_2}\right)$. Consequently,

$$\left\| \frac{v}{\|v\|_2} - \frac{g}{\|g\|_2} \right\|_2 \le \frac{\|\delta\|_2}{\|g\|_2} + \frac{|\|g\|_2 - \|v\|_2|}{\|g\|_2} \le \frac{2\|\delta\|_2}{\|g\|_2}.$$

Taking the inner product of the normalized key-direction difference with the unit query direction and applying Cauchy–Schwarz gives Eq. (30). Combining it with the exact gain from Theorem 5 and averaging over anchors gives Eq. (31). □

The mean coupling gain remains strictly positive when the ideal mean gain exceeds the relational perturbation penalty, namely $L^{-1}\sum_a \Gamma_a > 2L^{-1}\sum_a \epsilon_a$. This corollary characterizes the robustness of the common-core mechanism to perturbations of the relational trajectories.

D.3 Complementary Supports and Finite-Rank Correction of Private Residual Branches

D.3.1 Notation and Parameterization

We fix the input, layer, and attention head and analyze first-order corrections around a fixed shared representation in the linear content-score model, omitting layer and head indices. Let $X \in \mathbb{R}^{T \times d_{\mathrm{model}}}$ and $P_X = XX^\dagger$. The shared representation is $Q^S = ZB_Q$, $K^S = ZB_K$, with $\mathrm{col}(Z) \subseteq \mathrm{col}(X)$.

Let $S^\star$ be an input-realizable target linear content score, so $S^\star = P_X S^\star P_X$, and define $E = S^\star - Q^S(K^S)^\top$. Hence $E = P_X E P_X$. Set $P_Q = Q^S(Q^S)^\dagger$ and $P_K = K^S(K^S)^\dagger$.

After absorbing fixed nonzero branch-mixing scales into the activations, consider the paths $Q(t) = Q^S + t\delta Q$ and $K(t) = K^S + t\delta K$. The first-order score correction is

$$\delta S = \frac{\mathrm{d}}{\mathrm{d}t} Q(t)K(t)^\top \Big|_{t=0} = \delta Q (K^S)^\top + Q^S \delta K^\top. \tag{32}$$

The input-realizable correction set is $\mathcal{D}_X = \{D \in \mathbb{R}^{T \times d_{\mathrm{head}}} : P_X D = D\}$. A private branch $\delta Q = XA^{Qp}B^{Qp}$ corresponds exactly to a correction in $\mathcal{D}_X$ with rank at most r_q. Indeed, one direction follows from the parameterization; conversely, for any such δQ, take a rank factorization $\delta Q = UV$ with $\mathrm{col}(U) \subseteq \mathrm{col}(X)$ and set $A^{Qp} = X^\dagger U$, $B^{Qp} = V$. The key-private case is identical.

Decompose the residual into

$$\begin{aligned} E_{11} &= P_Q E P_K, & E_{10} &= P_Q E (I_T - P_K), \\ E_{01} &= (I_T - P_Q) E P_K, & E_{00} &= (I_T - P_Q) E (I_T - P_K). \end{aligned} \tag{33}$$

D.3.2 Theorem and Proof

Theorem 6 (Private-branch supports and finite-rank recovery)**.** *The four blocks in Eq. (33) are pairwise orthogonal in the Frobenius inner product. With sufficient private rank,*

$$\begin{aligned} \inf_{\delta Q \in \mathcal{D}_X} \|E - \delta Q (K^S)^\top\|_F^2 &= \|E_{10}\|_F^2 + \|E_{00}\|_F^2, \\ \inf_{\delta K \in \mathcal{D}_X} \|E - Q^S \delta K^\top\|_F^2 &= \|E_{01}\|_F^2 + \|E_{00}\|_F^2, \\ \inf_{\delta Q, \delta K \in \mathcal{D}_X} \|E - \delta Q (K^S)^\top - Q^S \delta K^\top\|_F^2 &= \|E_{00}\|_F^2. \end{aligned} \tag{34}$$

Let $\mathcal{E}_Q(r_q)$ and $\mathcal{E}_K(r_k)$ denote the corresponding single-sided optimal errors with rank constraints. Then

$$\mathcal{E}_Q(r_q) = \|E(I_T - P_K)\|_F^2 + \sum_{j > r_q} \sigma_j(EP_K)^2, \tag{35}$$

$$\mathcal{E}_K(r_k) = \|(I_T - P_Q)E\|_F^2 + \sum_{j > r_k} \sigma_j(P_Q E)^2. \tag{36}$$

For the joint finite-rank correction, let the optimal error be $\mathcal{E}_{QK}(r_q, r_k)$. Then

$$\|E_{00}\|_F^2 \le \mathcal{E}_{QK}(r_q, r_k) \le \|E_{00}\|_F^2 + \sum_{j > r_q} \sigma_j(EP_K)^2 + \sum_{j > r_k} \sigma_j(E_{10})^2. \tag{37}$$

Therefore, keeping the query-private rank fixed, the improvement obtained by adding key-private capacity satisfies

$$\mathcal{E}_Q(r_q) - \mathcal{E}_{QK}(r_q, r_k) \ge \sum_{j \le r_k} \sigma_j(E_{10})^2. \tag{38}$$

In particular, the improvement is strictly positive when $E_{10} \neq 0$ and $r_k \ge 1$.

Proof of Theorem 6. Using the decompositions $I_T = P_Q + (I_T - P_Q)$ and $I_T = P_K + (I_T - P_K)$ gives $E = E_{11} + E_{10} + E_{01} + E_{00}$. Distinct blocks have a Frobenius inner product containing either $P_Q(I_T - P_Q) = 0$ or $P_K(I_T - P_K) = 0$. Cyclicity of the trace therefore proves pairwise orthogonality.

For a query-private correction $R_Q = \delta Q (K^S)^\top$, the identity $(K^S)^\top (I_T - P_K) = 0$ gives $R_Q(I_T - P_K) = 0$. Hence

$$\|E - R_Q\|_F^2 = \|E(I_T - P_K)\|_F^2 + \|EP_K - R_Q\|_F^2.$$

Taking $\delta Q = EP_K((K^S)^\top)^\dagger$ realizes $R_Q = EP_K$. The columns of this correction lie in $\mathrm{col}(X)$, so the lower bound is attainable and the first equality in Eq. (34) follows.

The key-private result follows symmetrically. In particular, taking $\delta K^\top = (Q^S)^\dagger P_Q E$ realizes $Q^S \delta K^\top = P_Q E$ and gives the second equality.

For the joint correction,

$$(I_T - P_Q)\left[\delta Q (K^S)^\top + Q^S \delta K^\top\right](I_T - P_K) = 0.$$

Thus its error is at least $\|E_{00}\|_F^2$. Conversely, $E - E_{00} = EP_K + P_Q E(I_T - P_K)$. The first term is query-private reachable, while the second is realized by taking $\delta K^\top = (Q^S)^\dagger P_Q E(I_T - P_K)$. Since $E = P_X E P_X$ and both shared supports lie in $\mathrm{col}(X)$, these corrections belong to $\mathcal{D}_X$. The joint lower bound is therefore attained.

Now impose finite ranks. A query-side reachable score correction satisfies $RP_K = R$, $P_X R = R$, and $\mathrm{rank}(R) \le r_q$. Conversely, any R satisfying these conditions is realized by $\delta Q = R((K^S)^\top)^\dagger$. Thus the finite-rank problem is the rank-r_q approximation of EP_K. Its truncated SVD preserves the original

left and right supports and is therefore a valid correction, with residual error $\sum_{j>r_q}\sigma_j(EP_K)^2$. Adding the orthogonal term $\|E(I_T - P_K)\|_F^2$ proves Eq. (35). Applying the same argument to the transposed problem gives Eq. (36).

For the joint finite-rank construction, let R_Q be the rank-r_q truncated approximation of EP_K and let R_K be the rank-r_k truncated approximation of E_{10}. Both are realizable by the corresponding private branches, and

$$E - R_Q - R_K = E_{00} + (EP_K - R_Q) + (E_{10} - R_K).$$

The three terms on the right are pairwise orthogonal. Hence the construction has error $\|E_{00}\|_F^2 + \sum_{j>r_q}\sigma_j(EP_K)^2 + \sum_{j>r_k}\sigma_j(E_{10})^2$, which proves the upper bound in Eq. (37); the lower bound follows from the first-order inaccessibility of E_{00}.

Subtracting this joint upper bound from Eq. (35) gives

$$\mathcal{E}_Q(r_q) - \mathcal{E}_{QK}(r_q, r_k) \geq \|E_{10}\|_F^2 - \sum_{j>r_k}\sigma_j(E_{10})^2 = \sum_{j\leq r_k}\sigma_j(E_{10})^2.$$

This proves Eq. (38). □

Connection to the main-text Theorem 3. Theorem 6 gives a support-level explanation for the placement of private residual branches. Query-private correction acts against the shared key support, while key-private correction adds key-side score directions against the shared query support. The joint branch therefore opens complementary score-space capacity.

The finite-rank terms make the result directly relevant to the implemented parameterization. The query-private rank controls the truncation error of EP_K, the key-private rank controls the additional approximation of E_{10} in the joint construction, and the remaining E_{00} block records the first-order residual outside both shared supports. These results characterize the finite-rank approximation capacity of the private branches and inform capacity allocation; the specific allocation under a fixed total budget is assessed through ablation experiments.

For standard attention, a sufficiently expressive shared key support can make the additional key-side residual relatively small, so query-private capacity gives an efficient correction. In MLA-style compressed-KV attention, the cached content latent constrains the key support and can increase the energy of the E_{10} block. The theory therefore gives a positive structural reason for using joint query-private / key-private correction in the MLA regime.

D.3.3 Cross-Head Sharing of Private Input Factors

The private input factors in the main-text parameterization are shared across output heads. We extend the single-head analysis to this structure.

Let E_h denote the residual of head h, with corresponding projectors $P_{Q,h}$ and $P_{K,h}$. Define $\mathbb{E} = [E_1 \ \cdots \ E_H]$ and $\mathbb{P}_K = \mathrm{diag}(P_{K,1}, \ldots, P_{K,H})$.

Corollary 2 (Finite-rank recovery with shared private input factors)**.** *Suppose $[\delta Q_1 \ \cdots \ \delta Q_H] = XA^{Qp}[B_{Qp,1} \ \cdots \ B_{Qp,H}]$, where A^{Qp} has r_q columns. Then the optimal total query-private error is*

$$\mathcal{E}_Q^{(H)}(r_q) = \|\mathbb{E}(I_{HT} - \mathbb{P}_K)\|_F^2 + \sum_{j>r_q}\sigma_j(\mathbb{E}\mathbb{P}_K)^2. \tag{39}$$

Let $E_{10,h} = P_{Q,h}E_h(I_T - P_{K,h})$. If a cross-head shared key-private input factor with r_k columns is added, then

$$\mathcal{E}_Q^{(H)}(r_q) - \mathcal{E}_{QK}^{(H)}(r_q, r_k) \geq \max_{\substack{s_1,\ldots,s_H\in\mathbb{Z}_{\geq 0}\\ \sum_h s_h\leq r_k}} \sum_{h=1}^{H}\sum_{j\leq s_h}\sigma_j(E_{10,h})^2. \tag{40}$$

Thus, if at least one head has nonzero $E_{10,h}$, at least one key-private input direction gives a strict improvement in the optimal joint first-order error.

Proof of Corollary 2. Let $\mathbb{K} = \mathrm{diag}((K_1^S)^\top, \ldots, (K_H^S)^\top)$ and $\delta\mathbb{Q} = [\delta Q_1 \ \cdots \ \delta Q_H]$. The total query-private objective is $\|\mathbb{E} - \delta\mathbb{Q}\mathbb{K}\|_F^2$, and $\mathbb{K}^\dagger\mathbb{K} = \mathbb{P}_K$. As in Theorem 6, the optimal score

correction is the rank-r_q truncated approximation $R_\star$ of $\mathbb{E}\mathbb{P}_K$, realized by $\delta\mathbb{Q} = R_\star\mathbb{K}^\dagger$. Its rank is at most r_q and its column space lies in $\mathrm{col}(X)$, so it factors into a shared XA^{Qp} and head-specific readouts. This proves Eq. (39).

To prove Eq. (40), fix the optimal query correction. For any allocation with $\sum_h s_h \le r_k$, take in head h the rank-s_h truncated approximation of $E_{10,h}$ as the key-side score correction. It is Frobenius-orthogonal to the current query-reachable component and therefore reduces the squared error of head h by $\sum_{j\le s_h} \sigma_j(E_{10,h})^2$. The corresponding activation correction has rank at most s_h, and the concatenated correction has rank at most $\sum_h s_h \le r_k$. Its column space remains in $\mathrm{col}(X)$, so one shared rank-r_k private input factor realizes all head corrections. Summing the gains and maximizing over feasible allocations gives Eq. (40). □

This corollary connects the score-space theorem to the cross-head sharing used in the model. The input factor is shared across heads, while the readouts allocate the available private directions according to the residual spectra of individual heads. The resulting parameterization preserves cross-head efficiency and retains head-specific correction capacity.

D.3.4 Support Interpretation for an MLA Cached Latent

Proposition 2 (Query-private error under a fixed cached latent)**.** *Let $K_h^S = C^{KV}B_{K,h}$ and $P_C = C^{KV}(C^{KV})^\dagger$. With keys fixed, every query correction satisfies*

$$\sum_h \|E_h - \delta Q_h (K_h^S)^\top\|_F^2 \ge \sum_h \|E_h(I_T - P_C)\|_F^2. \tag{41}$$

For nested latent supports $\mathrm{col}(C_1) \subseteq \mathrm{col}(C_2)$,

$$\|E(I_T - P_{C_1})\|_F^2 = \|E(I_T - P_{C_2})\|_F^2 + \|E(P_{C_2} - P_{C_1})\|_F^2. \tag{42}$$

Proof of Proposition 2. Because $\mathrm{col}(K_h^S) \subseteq \mathrm{col}(C^{KV})$, we have $(K_h^S)^\top(I_T - P_C) = 0$. The right-projected residual is therefore always $E_h(I_T - P_C)$. Contractivity of orthogonal projection and summation over heads give Eq. (41).

Under nested supports, $P_{C_2} - P_{C_1}$ and $I_T - P_{C_2}$ are orthogonal projectors whose sum is $I_T - P_{C_1}$. The squared-norm orthogonal decomposition gives Eq. (42). □

Connection to the MLA placement result. Proposition 2 shows that the support of the cached latent directly determines the score-space directions accessible to query-private correction. Query-private capacity adjusts the query representation against the existing cached key support, while a key-private branch supplies additional key-side directions generated from the original input.

When the cached latent has a concentrated support, $E(I_T - P_C)$ measures the residual energy outside the shared key path. This gives a direct structural explanation for the additional value of key-private correction in MLA-style compressed-KV attention and connects the cached-latent geometry to the placement ablations in the main text.

The full two-branch score expansion is $(Q^S + \delta Q)(K^S + \delta K)^\top = Q^S(K^S)^\top + \delta Q(K^S)^\top + Q^S\delta K^\top + \delta Q\delta K^\top$. The preceding results characterize the first-order correction terms; the joint branch also supplies the second-order direction represented by $\delta Q\delta K^\top$. Thus, the shared trunk, query-private branch, and key-private branch form a score-aware decomposition with complementary roles.

E Additional Discussion

In this section, we present additional discussions for our key observation and efficient architecture design.

E.1 Relation to Existing Low-rank Training and QK Compression

QK-coupling uses an operator-specific empirical signal to guide a low-rank parameterization of the trainable QK path. The aligned anchor-relative trajectories motivate common routing coordinates, and the shared routing trunk represents these coordinates with separate Q/K readouts. Private residual branches provide complementary score-space capacity, yielding a rank-budgeted shared–private design.

This design complements methods targeting optimizer states, general adaptation subspaces, head sharing, and KV-cache compression. Its contribution is the connection between a QK-specific relational diagnostic, a trainable shared–private parameterization, and measured quality–efficiency outcomes across pre-training and fine-tuning.

E.2 Why Square-Root Mixing Coefficients?

The square-root coefficients provide a nominal allocation of branch energy. For $Q = \sqrt{\alpha}Q^S + \sqrt{1-\alpha}Q^P$,

$$\begin{aligned}\|Q\|_F^2 = \alpha\|Q^S\|_F^2 + (1-\alpha)\|Q^P\|_F^2 \\ + 2\sqrt{\alpha(1-\alpha)}\langle Q^S, Q^P\rangle_F.\end{aligned} \tag{43}$$

For equal-energy, orthogonal branches, the two contributions are exactly the fractions α and $1-\alpha$ of the total energy. Comparable branch energies and weak correlation give the corresponding approximate interpretation. The mixing coefficients therefore provide explicit control of branch coefficient scales, while Eq. (43) accounts for their interaction.

E.3 Organization of Empirical and Theoretical Sections

The intended logical order of our exposition is as follows: first, define the anchor-relative trajectory and multi-anchor QK-coupling score; second, examine the observed coupling across the evaluated dense, GQA, and MoE models; third, analyze how the common low-dimensional relational structure and coupling discrepancy control the error of a shared representation; and finally, study how core-preserving MLA-style compression can strengthen this coupling. Together with the score-space analysis of private corrections, these results motivate a shared routing trunk with separate Q/K output maps and lightweight private residual branches.

E.4 Numerical Example for Table 1

We give two concrete examples to make the cost formulas in Table 1 explicit. For a standard-attention 350M-scale setting with $d_{\text{model}} = d_{\text{qk}} = 1024, r_s = 128, r_q = 64$, the full QK parameter count per layer is $P_{\text{std}}^{QK} = 2d_{\text{model}}d_{\text{qk}} = 2 \times 1024^2 = 2.10$M. The QK-coupled parameter count is: $P_{\text{qkc}}^{QK} = d_{\text{model}}r_s + 2r_s d_{\text{qk}} + d_{\text{model}}r_q + r_q d_{\text{qk}} = 1024\times128 + 2\times128\times1024 + 1024\times64 + 64\times1024 = 0.52$M. Thus the QK operator-level parameter, optimizer-state, and projection-FLOP saving is 75%.

For the LoRA-style fine-tuning example on RoBERTa-Base ($L = 12, d_{\text{model}} = d_{\text{qk}} = 768, r = 16, r_s = 8, r_q = 4$), independent rank-$r$ Q/K adapters use $2Lr(d_{\text{model}} + d_{\text{qk}}) = 0.59$M QK adapter parameters. QK-coupling uses $L\big(d_{\text{model}}r_s + 2r_s d_{\text{qk}} + d_{\text{model}}r_q + r_q d_{\text{qk}}\big) = 0.29$M, matching the parameter accounting reported in our experiments.

E.5 Reading Guide for the Theoretical Results

- **Theorem 1** relates multi-anchor QK-coupling to shared low-dimensional relational representations. The shared reconstruction error is controlled by the common component's spectral tail and the Q/K relational mismatch. This motivates a shared low-rank routing trunk with separate Q/K output maps.
- **Theorem 2** studies how core-preserving MLA-style compression strengthens multi-anchor QK-coupling. Under the stated common-core separation and spectral-dominance conditions, retaining the common KV core while removing nonzero key-specific relational residuals increases coupling when the relational components are mutually orthogonal. This motivates exploiting the common content representation in the MLA instantiation.

- **Theorem 3** characterizes complementary first-order score-space corrections around a fixed shared representation. Query-private and key-private branches recover different reachable supports, with finite private ranks controlling approximation within those supports. This motivates lightweight private capacity alongside the shared trunk.

E.6 Intuition Behind Private Residual Branches

The private residual branch is needed because attention scores are bilinear in Q and K. A shared routing trunk can represent score components of the form $Q^S K^{S\top}$, which captures the part of the score matrix explained by the common Q/K routing geometry. After this shared component is fixed, we analyze the remaining score-space residual through first-order query-private and key-private corrections.

Query-private correction adds score directions of the form $Q^P K^{S\top}$ (query moves while key stays in the shared subspace). Key-private correction adds directions of the form $Q^S K^{P\top}$ (key moves while query stays). These two corrections are complementary. In standard attention, query-private correction is often enough. In MLA-style compressed-KV attention, the key content path is already compressed, so joint query-private / key-private correction is often more useful.

E.7 Why Use a Shared Routing Trunk with Private Residual Branches?

The shared-private structure is motivated by the geometry suggested by multi-anchor QK-coupling. Representing the common component twice, as in independent low-rank Q and K factorizations, duplicates the shared routing geometry. A shared routing trunk avoids this duplication by learning the common low-dimensional routing coordinates once. Private residual branches extend the reachable score-space directions while retaining a shared routing component. Therefore, the proposed structure is best understood as a rank-budgeted decomposition.

E.8 Motivation for the Anchor-Relative QK-coupling Score

The anchor-relative score measures how Q and K organize cross-layer relationships with respect to the same anchor-layer indices. It compares Q-to-Q and K-to-K trajectories in their native projected spaces and summarizes their directional agreement. This operator-specific diagnostic motivates the shared-routing hypothesis evaluated by the proposed parameterization and the accompanying experiments.

Interpreting the magnitude of coupling. For each anchor a, let $\widehat{u}_A^{(a)} = u_A^{(a)}/\|u_A^{(a)}\|_2$ denote the unit-normalized, nonzero trajectory for $A \in \{Q, K\}$, with the anchor self-similarity excluded. Since both normalized trajectories have unit norm and their inner product equals $\kappa_{QK}(a)$,

$$\begin{aligned}\frac{1}{L}\sum_{a=1}^{L}\left\|\widehat{u}_Q^{(a)} - \widehat{u}_K^{(a)}\right\|_2^2 &= \frac{1}{L}\sum_{a=1}^{L}\left(2 - 2\kappa_{QK}(a)\right)\\ &= 2\left(1 - \bar{\kappa}_{QK}\right).\end{aligned}$$

Orthogonal unit trajectories have squared distance 2, providing a geometric reference for interpreting the score. The Qwen2.5-3B Base value $\bar{\kappa}_{QK} = 0.338$ reported in Section 3.2 therefore corresponds to a mean squared distance of 1.324, a 33.8% reduction relative to this reference. Thus, the observed value represents nontrivial positive relational alignment. This common relational structure motivates shared routing coordinates, while independent output maps and private branches retain operator-specific flexibility.

F Storage, FLOPs, and KV-Cache Analysis

In this section, we analyze the storage, computation, and KV-cache cost of the QK-coupled parameterization. All costs are reported per Transformer layer unless otherwise specified. We use the notation in Table 5: d_{model} is the hidden dimension, d_{qk} is the total Q/K output dimension, d_v is the total value dimension, B is the batch size, T is the sequence length, and $N = BT$ is the number of

training tokens in one batch. The formulas using one d_{qk} describe equal-width Q/K paths. Native unequal-width Q/K paths use the following matrix-parameter counts:

$$\begin{aligned} P_{\mathrm{std}}^{QK} &= d_{\mathrm{model}}(d_Q + d_K), \\ P_{\mathrm{qkc}}^{QK} &= d_{\mathrm{model}} r_s + r_s(d_Q + d_K) \\ &\quad + r_q(d_{\mathrm{model}} + d_Q) + r_k(d_{\mathrm{model}} + d_K). \end{aligned} \tag{44}$$

For independent rank-r Q/K adapters, the count is $r(2d_{\mathrm{model}}+d_Q+d_K)$. The equal-width expressions follow by setting $d_Q = d_K = d_{\mathrm{qk}}$; the square case additionally sets $d_{\mathrm{qk}} = d_{\mathrm{model}}$.

For projection-FLOP accounting, the batch and sequence dimensions are flattened, so $X_l \in \mathbb{R}^{N \times d_{\mathrm{model}}}$ with $N = BT$. The corresponding activation matrices use N rows; the main-text formulas are written for a single sequence with T rows.

For a matrix multiplication between an $n \times a$ matrix and an $a \times b$ matrix, we count

$$\mathrm{FLOPs}\big((n \times a)(a \times b)\big) = 2nab. \tag{45}$$

For training FLOPs of a linear projection, we count the forward pass, the input gradient, and the weight gradient. Thus,

$$\mathrm{FLOPs}_{\mathrm{train}}(XW) = 3\,\mathrm{FLOPs}_{\mathrm{fwd}}(XW). \tag{46}$$

Element-wise scaling and addition introduced by private residual branches are lower-order $O(Nd_{\mathrm{qk}})$ terms and are omitted from the matrix-multiplication FLOP count. For Adam-style training storage, we count four copies of trainable parameters: parameter, gradient, first moment, and second moment.

F.1 Standard Attention Pre-Training

Standard Q/K projections. In standard attention, the query and key projections are two independent matrices:

$$W_l^Q, W_l^K \in \mathbb{R}^{d_{\mathrm{model}} \times d_{\mathrm{qk}}}. \tag{47}$$

The number of trainable Q/K parameters is

$$P_{\mathrm{std}}^{QK} = 2d_{\mathrm{model}}d_{\mathrm{qk}}. \tag{48}$$

With Adam, the corresponding training storage is

$$M_{\mathrm{std}}^{QK} = 4P_{\mathrm{std}}^{QK} = 8d_{\mathrm{model}}d_{\mathrm{qk}}. \tag{49}$$

The forward FLOPs for Q/K projection are

$$F_{\mathrm{std,fwd}}^{QK} = 4Nd_{\mathrm{model}}d_{\mathrm{qk}}, \tag{50}$$

and the training FLOPs are

$$F_{\mathrm{std,train}}^{QK} = 3F_{\mathrm{std,fwd}}^{QK} = 12Nd_{\mathrm{model}}d_{\mathrm{qk}}. \tag{51}$$

QK-coupled parameterization. For standard attention pre-training, the QK-coupled parameterization uses a shared routing rank r_s and a query-private rank r_q. The shared routing trunk and query-private correction are defined as

$$Z_l = X_l A_l^S, \qquad Q_l^S = Z_l B_l^Q, \qquad K_l^S = Z_l B_l^K, \qquad Q_l^P = X_l A_l^{Qp} B_l^{Qp}. \tag{52}$$

The trainable matrices have sizes

$$\begin{aligned} &A_l^S \in \mathbb{R}^{d_{\mathrm{model}} \times r_s}, \quad B_l^Q \in \mathbb{R}^{r_s \times d_{\mathrm{qk}}}, \quad B_l^K \in \mathbb{R}^{r_s \times d_{\mathrm{qk}}}, \\ &A_l^{Qp} \in \mathbb{R}^{d_{\mathrm{model}} \times r_q}, \quad B_l^{Qp} \in \mathbb{R}^{r_q \times d_{\mathrm{qk}}}. \end{aligned} \tag{53}$$

Ignoring the negligible per-head scaling factors, the QK-coupled parameterization has parameter cost

$$P_{\mathrm{qkc}}^{QK} = d_{\mathrm{model}} r_s + 2r_s d_{\mathrm{qk}} + d_{\mathrm{model}} r_q + r_q d_{\mathrm{qk}}. \tag{54}$$

When $d_{\text{qk}} = d_{\text{model}}$, Eq. (54) reduces to $d_{\text{model}}(3r_s + 2r_q)$. If H learnable per-head query scaling factors are counted, the exact count becomes $P_{\text{qkc}}^{QK} + H$. With Adam, the dominant training storage is

$$M_{\text{qkc}}^{QK} = 4P_{\text{qkc}}^{QK}. \tag{55}$$

The parameter and Adam-memory saving ratios against standard Q/K projections are

$$\text{Saving}_{\text{param}} = \text{Saving}_{\text{Adam}} = 1 - \frac{P_{\text{qkc}}^{QK}}{P_{\text{std}}^{QK}} = 1 - \frac{d_{\text{model}} r_s + 2r_s d_{\text{qk}} + d_{\text{model}} r_q + r_q d_{\text{qk}}}{2d_{\text{model}} d_{\text{qk}}}. \tag{56}$$

In the square case, this becomes $1 - (3r_s + 2r_q)/(2d_{\text{model}})$.

The forward FLOPs are obtained from each matrix multiplication:

$$\begin{aligned} F_{\text{qkc,fwd}}^{QK} &= \underbrace{2Nd_{\text{model}} r_s}_{X_l A_l^S} + \underbrace{2Nr_s d_{\text{qk}}}_{Z_l B_l^Q} + \underbrace{2Nr_s d_{\text{qk}}}_{Z_l B_l^K} + \underbrace{2Nd_{\text{model}} r_q}_{X_l A_l^{Q_p}} + \underbrace{2Nr_q d_{\text{qk}}}_{(X_l A_l^{Q_p}) B_l^{Q_p}} \\ &= 2NP_{\text{qkc}}^{QK}. \end{aligned} \tag{57}$$

The corresponding training FLOPs are

$$F_{\text{qkc,train}}^{QK} = 3F_{\text{qkc,fwd}}^{QK} = 6NP_{\text{qkc}}^{QK}. \tag{58}$$

Therefore, the QK projection FLOP saving ratio is

$$\text{Saving}_{\text{FLOP}} = 1 - \frac{F_{\text{qkc,fwd}}^{QK}}{F_{\text{std,fwd}}^{QK}} = 1 - \frac{P_{\text{qkc}}^{QK}}{P_{\text{std}}^{QK}}. \tag{59}$$

The same ratio holds for training FLOPs because both standard and QK-coupled linear projections are multiplied by the same factor of three.

F.2 MLA-Style Compressed-KV Attention Pre-Training

For the SmolLM-style baseline in Table 12, the QK projection count includes the query projection, the positional-key projection, the joint KV down-projection, and the content-key up-projection. The value up-projection is accounted for separately.

We next analyze the MLA-style compressed-KV attention setting. The baseline cached content latent has dimension d_c, the QK-coupled shared cached content latent has dimension d_s, the cached key-private latent has dimension d_{kp}, the positional key-cache dimension is d_{pos}, the total content Q/K output dimension is d_{qk}, and the total value dimension is d_v.

Baseline MLA-style compressed-KV attention content paths. We first give the direct-query MLA-style compressed-KV attention accounting; the compressed-query variant is given in Eq. (63). The direct-query content path computes

$$C_l^{KV} = X_l W_l^{DKV}, \quad K_l^C = C_l^{KV} W_l^{UK}, \quad V_l = C_l^{KV} W_l^{UV}, \quad Q_l^C = X_l W_l^{Q,C}. \tag{60}$$

The corresponding matrix shapes are

$$W_l^{DKV} \in \mathbb{R}^{d_{\text{model}} \times d_c}, \quad W_l^{UK} \in \mathbb{R}^{d_c \times d_{\text{qk}}}, \quad W_l^{UV} \in \mathbb{R}^{d_c \times d_v}, \quad W_l^{Q,C} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{qk}}}. \tag{61}$$

Thus the content-path parameter count is

$$P_{\text{std}}^{\text{MLA}} = d_{\text{model}} d_c + d_c d_{\text{qk}} + d_c d_v + d_{\text{model}} d_{\text{qk}}. \tag{62}$$

For a query latent of rank r_Q^{lat}, the query-branch parameter count is $d_{\text{model}} r_Q^{\text{lat}} + r_Q^{\text{lat}} d_{\text{qk}}$. The resulting content-path count is

$$P_{\text{std,latent}}^{\text{MLA}} = d_{\text{model}} d_c + d_c(d_{\text{qk}} + d_v) + d_{\text{model}} r_Q^{\text{lat}} + r_Q^{\text{lat}} d_{\text{qk}}. \tag{63}$$

This expression accounts for the compressed-query variant introduced in Eq. (4); for this variant, use $P_{\text{std,latent}}^{\text{MLA}}$ in the baseline storage and projection-FLOP counts below. The Adam storage is

$$M_{\text{std}}^{\text{MLA}} = 4P_{\text{std}}^{\text{MLA}}. \tag{64}$$

The forward FLOPs of the content branches are

$$\begin{aligned} F^{\mathrm{MLA}}_{\mathrm{std,fwd}} &= \underbrace{2Nd_{\text{model}}d_c}_{X_l W_l^{DKV}} + \underbrace{2Nd_c d_{\text{qk}}}_{C_l^{KV} W_l^{UK}} + \underbrace{2Nd_c d_v}_{C_l^{KV} W_l^{UV}} + \underbrace{2Nd_{\text{model}}d_{\text{qk}}}_{X_l W_l^{Q,C}} \\ &= 2NP^{\mathrm{MLA}}_{\mathrm{std}}. \end{aligned} \tag{65}$$

The training FLOPs are

$$F^{\mathrm{MLA}}_{\mathrm{std,train}} = 3F^{\mathrm{MLA}}_{\mathrm{std,fwd}}. \tag{66}$$

During decoding, MLA-style compressed-KV attention stores the compressed content latent and the positional-key branch. Thus, the baseline per-token, per-layer KV cache is

$$C^{\mathrm{KV-cache}}_{\mathrm{std}} = d_c + d_{\text{pos}}. \tag{67}$$

QK-coupled variant of MLA-style compressed-KV attention. In the QK-coupled variant of MLA-style compressed-KV attention, we use a smaller cached shared content latent $C_l^S \in \mathbb{R}^{N\times d_s}$, a shared routing rank r_s, a query-private rank r_q, and a cached key-private latent dimension d_{kp}. Here $A_l^S \in \mathbb{R}^{d_s\times r_s}$, and the input-generated key-private factor uses $r_k = d_{kp}$. The content paths are

$$\begin{aligned} C_l^S &= X_l W_l^S, & V_l &= C_l^S W_l^{UV,S}, & Z_l &= C_l^S A_l^S, \\ Q_l^{C,S} &= Z_l B_l^Q, & K_l^{C,S} &= Z_l B_l^K, & & \\ Q_l^{C,P} &= X_l A_l^{Qp} B_l^{Qp}, & C_l^{Kp} &= X_l A_l^{Kp}, & K_l^{C,P} &= C_l^{Kp} B_l^{Kp}. \end{aligned} \tag{68}$$

The corresponding parameter count is

$$\begin{aligned} P^{\mathrm{MLA}}_{\mathrm{qkc}} = &\underbrace{d_{\text{model}}d_s}_{X_l\to C_l^S} + \underbrace{d_s d_v}_{C_l^S\to V_l} + \underbrace{d_s r_s}_{C_l^S\to Z_l} + \underbrace{2r_s d_{\text{qk}}}_{Z_l\to Q_l^C, K_l^C} + \underbrace{d_{\text{model}}r_q + r_q d_{\text{qk}}}_{Q\text{-private}} \\ &+ \underbrace{d_{\text{model}}d_{kp} + d_{kp}d_{\text{qk}}}_{K\text{-private}}. \end{aligned} \tag{69}$$

If the per-head query and key scaling factors are included, the exact count is $P^{\mathrm{MLA}}_{\mathrm{qkc}} + 2H$. The Adam storage is

$$M^{\mathrm{MLA}}_{\mathrm{qkc}} = 4P^{\mathrm{MLA}}_{\mathrm{qkc}}. \tag{70}$$

The parameter and Adam-memory saving ratios are

$$\mathrm{Saving}^{\mathrm{MLA}}_{\mathrm{param}} = \mathrm{Saving}^{\mathrm{MLA}}_{\mathrm{Adam}} = 1 - \frac{P^{\mathrm{MLA}}_{\mathrm{qkc}}}{P^{\mathrm{MLA}}_{\mathrm{std}}}. \tag{71}$$

The forward FLOPs are

$$\begin{aligned} F^{\mathrm{MLA}}_{\mathrm{qkc,fwd}} = &\underbrace{2Nd_{\text{model}}d_s}_{X_l W_l^S} + \underbrace{2Nd_s d_v}_{C_l^S W_l^{UV,S}} + \underbrace{2Nd_s r_s}_{C_l^S A_l^S} + \underbrace{4Nr_s d_{\text{qk}}}_{Z_l B_l^Q, Z_l B_l^K} \\ &+ \underbrace{2Nd_{\text{model}}r_q + 2Nr_q d_{\text{qk}}}_{Q\text{-private}} + \underbrace{2Nd_{\text{model}}d_{kp} + 2Nd_{kp}d_{\text{qk}}}_{K\text{-private}} \\ = &\ 2NP^{\mathrm{MLA}}_{\mathrm{qkc}}. \end{aligned} \tag{72}$$

The training FLOPs are

$$F^{\mathrm{MLA}}_{\mathrm{qkc,train}} = 3F^{\mathrm{MLA}}_{\mathrm{qkc,fwd}}. \tag{73}$$

The content-path FLOP saving ratio is

$$\mathrm{Saving}^{\mathrm{MLA}}_{\mathrm{FLOP}} = 1 - \frac{F^{\mathrm{MLA}}_{\mathrm{qkc,fwd}}}{F^{\mathrm{MLA}}_{\mathrm{std,fwd}}} = 1 - \frac{P^{\mathrm{MLA}}_{\mathrm{qkc}}}{P^{\mathrm{MLA}}_{\mathrm{std}}}. \tag{74}$$

The query-private correction is computed for the current token, while the cache stores the shared content latent, the enabled key-private latent, and the positional-key branch. If the QK-coupled

positional-key dimension is d'_{pos}, the cache of the QK-coupled variant of MLA-style compressed-KV attention is

$$C_{\text{qkc}}^{\text{KV-cache}} = d_s + d_{kp} + d'_{\text{pos}}. \tag{75}$$

Therefore, the per-token, per-layer KV-cache saving in elements is

$$\Delta_{\text{KV}} = C_{\text{std}}^{\text{KV-cache}} - C_{\text{qkc}}^{\text{KV-cache}} = d_c + d_{\text{pos}} - (d_s + d_{kp} + d'_{\text{pos}}). \tag{76}$$

For a decoding context of length T over L cached layers, the total KV-cache saving is

$$\Delta_{\text{KV,total}} = TL\big[d_c + d_{\text{pos}} - (d_s + d_{kp} + d'_{\text{pos}})\big] \tag{77}$$

elements. If the cache uses b_{elem} bytes per element, the byte saving is

$$\Delta_{\text{KV,bytes}} = b_{\text{elem}} TL\big[d_c + d_{\text{pos}} - (d_s + d_{kp} + d'_{\text{pos}})\big]. \tag{78}$$

Thus, the QK-coupled variant of MLA-style compressed-KV attention yields additional KV-cache savings whenever

$$d_s + d_{kp} + d'_{\text{pos}} < d_c + d_{\text{pos}}. \tag{79}$$

F.3 LoRA-Style Fine-Tuning

Finally, we analyze the QK-coupled parameterization for LoRA-style fine-tuning. Here the base weights are frozen and only the low-rank updates are trainable. We count the additional adapter parameters and adapter FLOPs; the frozen base Q/K projection cost is the same across LoRA baselines and cancels in comparison.

Independent Q/K LoRA. For independent rank-r LoRA updates on both Q and K,

$$\begin{aligned} \Delta W_l^Q &= A_l^Q B_l^Q, \quad A_l^Q \in \mathbb{R}^{d_{\text{model}} \times r}, \quad B_l^Q \in \mathbb{R}^{r \times d_{\text{qk}}}, \\ \Delta W_l^K &= A_l^K B_l^K, \quad A_l^K \in \mathbb{R}^{d_{\text{model}} \times r}, \quad B_l^K \in \mathbb{R}^{r \times d_{\text{qk}}}. \end{aligned} \tag{80}$$

The trainable adapter parameter count is

$$P_{\text{std}}^{\text{LoRA-QK}} = 2r(d_{\text{model}} + d_{\text{qk}}). \tag{81}$$

The Adam storage for trainable adapters is

$$M_{\text{std}}^{\text{LoRA-QK}} = 8r(d_{\text{model}} + d_{\text{qk}}). \tag{82}$$

The forward adapter FLOPs are

$$F_{\text{std,fwd}}^{\text{LoRA-QK}} = 4Nr(d_{\text{model}} + d_{\text{qk}}), \tag{83}$$

and the training adapter FLOPs are

$$F_{\text{std,train}}^{\text{LoRA-QK}} = 12Nr(d_{\text{model}} + d_{\text{qk}}). \tag{84}$$

QK-coupled LoRA. QK-coupled LoRA uses a shared routing trunk for the Q/K update subspace and keeps query-private correction:

$$\Delta W_l^Q = \sqrt{\alpha_l^Q} A_l^S B_l^Q + \sqrt{1 - \alpha_l^Q} A_l^{Qp} B_l^{Qp}, \qquad \Delta W_l^K = A_l^S B_l^K. \tag{85}$$

The trainable parameter count is

$$P_{\text{qkc}}^{\text{LoRA-QK}} = d_{\text{model}} r_s + 2r_s d_{\text{qk}} + d_{\text{model}} r_q + r_q d_{\text{qk}}, \tag{86}$$

up to the negligible per-head scaling factors. The Adam storage is

$$M_{\text{qkc}}^{\text{LoRA-QK}} = 4P_{\text{qkc}}^{\text{LoRA-QK}}. \tag{87}$$

The adapter parameter and Adam-memory saving ratios are

$$\text{Saving}_{\text{param}}^{\text{LoRA}} = \text{Saving}_{\text{Adam}}^{\text{LoRA}} = 1 - \frac{P_{\text{qkc}}^{\text{LoRA-QK}}}{2r(d_{\text{model}} + d_{\text{qk}})}. \tag{88}$$

The forward adapter FLOPs are

$$F_{\text{qkc,fwd}}^{\text{LoRA}-\text{QK}} = \underbrace{2Nd_{\text{model}}r_s}_{X_l A_l^S} + \underbrace{2Nr_s d_{\text{qk}}}_{(X_l A_l^S)B_l^Q} + \underbrace{2Nr_s d_{\text{qk}}}_{(X_l A_l^S)B_l^K} + \underbrace{2Nd_{\text{model}}r_q}_{X_l A_l^{Qp}} + \underbrace{2Nr_q d_{\text{qk}}}_{(X_l A_l^{Qp})B_l^{Qp}} \tag{89}$$
$$= 2NP_{\text{qkc}}^{\text{LoRA}-\text{QK}}.$$

The training adapter FLOPs are

$$F_{\text{qkc,train}}^{\text{LoRA}-\text{QK}} = 6NP_{\text{qkc}}^{\text{LoRA}-\text{QK}}. \tag{90}$$

Compared with independent rank-r Q/K LoRA, the adapter FLOP saving ratio is

$$\text{Saving}_{\text{FLOP}}^{\text{LoRA}} = 1 - \frac{F_{\text{qkc,fwd}}^{\text{LoRA}-\text{QK}}}{F_{\text{std,fwd}}^{\text{LoRA}-\text{QK}}} = 1 - \frac{P_{\text{qkc}}^{\text{LoRA}-\text{QK}}}{2r(d_{\text{model}} + d_{\text{qk}})}. \tag{91}$$