SOFTMAX ATTENTION IS AN EXACT GRADIENT STEP: PREDICTABLE FINITE RESPONSES FROM ROPE DERIVATIVES

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformers are said to learn in context by gradient descent, but the claim is proven only where pretrained models do not operate: in linear attention, or at weights constructed to make it true. For the softmax attention they use, it survives only as an approximation that linearises the exponential, and whether pretrained models implement gradient descent at all is disputed (Shen et al., 2024). We show that no approximation is needed: every softmax attention head, for any weights, context and query, computes exactly one step of gradient descent. Its output is the uniform average of its values plus the prediction, at its query, of a linear model trained by one gradient step of weighted least-squares regression from keys to centred values. Each example is weighted by the exponential's divided difference $\varphi_1(x) = (e^x - 1)/x$ at its attention score, so every weight is positive and rises with the score. As scores shrink, the weights become uniform and the step becomes the linear-attention gradient step of von Oswald et al. (2023). The loss's curvature vanishes exactly on query edits that leave attention unchanged, and one linear solve finds the smallest edit realising requested attention ratios. The same identity, taken between two states of the head, gives the exact response to any finite edit, including moving a token, whose rotation follows from integrating RoPE's derivative. In pretrained Qwen2.5 and SmolLM2, the gradient step reproduces every evaluated head to floating-point precision, and across 153,600 executed positional edits, exact responses cut the error of linearised predictions by 73.6–98.3%. Softmax attention does not approximate gradient descent in context. It performs it, exactly, in every head.

1 INTRODUCTION

Language is organised hierarchically: words form phrases and phrases form sentences, often across long distances. Transformers assemble this structure with attention (Vaswani et al., 2017; Clark et al., 2019): each token reads earlier tokens through their keys, which set how much attention each gets, and their values, which set what it passes on.

Theory casts this read as one step of gradient descent on the context, exactly for linear attention (von Oswald et al., 2023) and approximately for softmax (Dai et al., 2023). Practice edits it: activation patching swaps keys and values in from another prompt (Zhang & Nanda, 2024), attention steering shifts weight to chosen tokens (Zhang et al., 2024), KV-cache compression evicts entries (Li et al., 2024), and under RoPE (Su et al., 2021) even moving a token edits its key.

Attribution patching (AtP) tried to predict their effect without running the model (Syed et al., 2024; Kramár et al., 2024). It failed because, without an exact calculus for finite attention edits, it could only linearise. But attention edits are rarely small: a patch replaces keys and values outright, a 16-position shift turns RoPE's fastest rotation more than twice around, and softmax is exponential. On Qwen2.5 key–value patches, AtP's error is at least a third of the effect, and sometimes exceeds it. Its successor AtP* recomputes the softmax for new keys but still reads new values through old attention weights, and at one layer of Qwen2.5-1.5B does worse than predicting no change. Adding an exact calculus, and nothing else, removes over 90% of AtP's error and cuts AtP*'s at least sixfold.

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

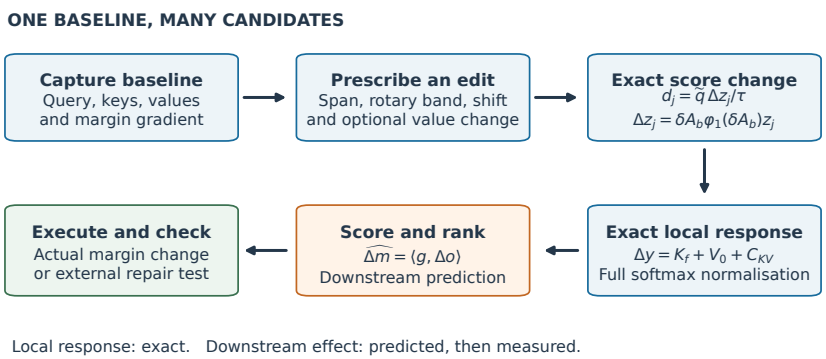

Figure 1: **Construct, score, and check an attention intervention.** The generator specifies a finite key displacement, and exact softmax normalisation gives the local response, including value changes. A shared gradient scores candidates; execution checks selected edits.

Theory and practice lack the same thing: an exact calculus of attention. We build one from a single function, the exponential divided difference $\varphi_1(x) = (e^x - 1)/x$ (Hochbruck & Ostermann, 2010):

$$\begin{aligned} &\text{moving a key:} && z(p+\delta) - z(p) = \delta A\, \varphi_1(\delta A)\, z(p), \\ &\text{shifting attention:} && p'_j - p_j = p_j\, t_j\, \varphi_1(t_j), \qquad t_j = d_j - \log \textstyle\sum_k p_k e^{d_k}, \\ &\text{reading the context:} && y - \mu = \textstyle\sum_j c_j\, s_j\, (v_j - \mu), \qquad c_j = \varphi_1(s_j)/Z > 0. \end{aligned} \tag{1}$$

The first integrates RoPE's derivative $\partial_p z = Az$; the second carries any score change d_j through softmax. The third is exact gradient descent: for any query and weights, a RoPE–softmax head takes one gradient step on a least-squares fit of its centred values to its keys, weighted by c_j. To first order, the three reduce to the tangent step, the linear edit score and linear-attention gradient descent.

We summarise our contributions as follows:

- **An exact calculus of attention.** Three φ_1 identities give a head's exact response to any finite key, score or value edit (Theorem 1), so one backward pass scores every candidate (Figure 1); a KL bound limits the error of first-order attribution, and endpoint secants factorise the whole network's logit change exactly (Appendix D.8).
- **Softmax attention is exact gradient descent.** The third identity holds for every RoPE–softmax head, query and weights (Section 3.3); every pretrained head we reconstruct obeys it (Appendix E).
- **An exact key–value interaction.** When keys and values are edited together, the exact response splits into a key term, a value term and their interaction $C_{KV} = \sum_j (p'_j - p_j)\varepsilon_j$, which tracks the executed interaction with $r > 0.99$. Across 162,304 executed edits in Qwen2.5 and SmolLM2, exact responses cut the positional Jacobian's error by 74–98%, and dropping C_{KV} raises the error on joint edits at least sixfold (Section 5).
- **Control and caching with guarantees.** Attention-preserving query edits are exactly the kernel of the least-squares curvature; one linear solve gives the smallest edit realising requested attention ratios or certifies that none exists; exact deletion effects rank cache entries; and query-set weights can beat any fixed affine predictor (Theorem 2).

2 From the RoPE derivative to a finite attention response

2.1 Integrating the positional generator

For fixed, unscaled RoPE frequencies, write

$$R(p) = e^{pA}, \qquad A = \operatorname{diag}_{r=1}^{d/2} \begin{pmatrix} 0 & -\omega_r \\ \omega_r & 0 \end{pmatrix}, \qquad z(p) = R(p) z_0. \tag{2}$$

Position is extended continuously for differentiation; integer positions are the usual evaluation points. Holding z_0 fixed gives $\partial_p^n z(p) = A^n z(p)$ for every $n \geq 1$. The fundamental theorem of calculus

now gives

$$z(p+\delta) - z(p) = \delta \int_0^1 A e^{t\delta A} z(p)\,\mathrm{d}t \\ = \delta A \varphi_1(\delta A) z(p), \qquad \varphi_1(B) = \sum_{n=0}^{\infty} \frac{B^n}{(n+1)!}. \tag{3}$$

This entire function covers zero-frequency blocks without an inverse of A. It also makes the approximation explicit: $\delta A z(p)$ is the first-order term, while $\varphi_1(\delta A)$ retains all higher orders. On plane r, the relevant scale is $\theta_r = \delta\omega_r$. The exact rotary displacement is bounded by $2\,\|z_r\|_2$, whereas the tangent has norm $|\theta_r|\,\|z_r\|_2$. A token shift can therefore be small in one band and far outside the tangent regime in another (Figure A.1, Appendix A).

2.2 Propagating the finite shift through softmax

Fix a deterministic attention head, its query, and a nonempty attended set. Using row vectors, let $\widetilde{q} = qR(p_i)^\top$, $\widetilde{k}_j = k_j R(p_j)^\top$, $\tau = \sqrt{d}$, $s_j = \widetilde{q}\widetilde{k}_j^\top/\tau$, $p_j = e^{s_j}/\sum_k e^{s_k}$, and $y = \sum_j p_j v_j$. Let A_b retain the selected rotary blocks and be zero elsewhere. Equation (3) supplies the score increment for a key shifted by δ in that band:

$$\widetilde{k}'_j = \widetilde{k}_j e^{\delta A_b^\top}, \qquad d_j = s'_j - s_j = \frac{\widetilde{q}\,\delta A_b \varphi_1(\delta A_b)\widetilde{k}_j^\top}{\tau}. \tag{4}$$

Unedited keys have $d_j = 0$. Define

$$D = \sum_j p_j e^{d_j}, \qquad t_j = d_j - \log D, \qquad p'_j = p_j e^{t_j}, \qquad p'_j - p_j = p_j t_j \varphi_1(t_j). \tag{5}$$

The scalar identity $e^t - 1 = t\varphi_1(t)$ therefore retains the exact softmax renormalisation.

Theorem 1 (Exact local response to finite edits)**.** *For the fixed query and attended set above, let $v'_j = v_j + \varepsilon_j$. For any finite score increments d_j,*

$$\boxed{\Delta y = y' - y = \sum_j p_j t_j \varphi_1(t_j)(v_j - y) + \sum_j p'_j \varepsilon_j.} \tag{6}$$

For a positional intervention, inserting (4) *gives the exact response of the full RoPE–softmax head.*

Proof. Subtract the two readouts and write $\Delta y = \sum_j (p'_j - p_j) v_j + \sum_j p'_j \varepsilon_j$. Since the probability changes sum to zero, replace v_j by $v_j - y$ in the first sum and use (5). □

Theorem 1 integrates rather than linearises: $p_j\varphi_1(t_j) = \int_0^1 p_j^{1-r} p_j'^{r}\,\mathrm{d}r$ averages each weight over the whole displacement rather than its tangent, as the exact chord follows the rotary circle (Figure A.1). Its first term rests on conservation: probabilities sum to one, so mass is only reallocated and each value is centred at y, while subtracting $\log D$ removes the common score shift that softmax ignores, its gauge symmetry. Its second term reads value edits at p'_j, so beyond V_0 it contains exactly the interaction C_{KV} of (8), which separate key and value attributions lose. Content stays fixed at the readout; a four-cell decomposition separates the upstream contextual changes of a reordered prompt (Appendix G).

2.3 The tangent limit and coupling between bands

For key-only edits, the positional Jacobian substitutes $d_j^{\mathrm{lin}} = \widetilde{q}\,\delta A_b \widetilde{k}_j^\top/\tau$ and predicts $\sum_j p_j d_j^{\mathrm{lin}}(v_j - y)$. Its local discrepancy has two explicit parts:

$$\Delta y - \Delta y_{\mathrm{Jac}} = \underbrace{\sum_j p_j t_j \big(\varphi_1(t_j) - 1\big)(v_j - y)}_{\text{softmax remainder}} + \underbrace{\sum_j p_j (d_j - d_j^{\mathrm{lin}})(v_j - y)}_{\text{rotary remainder}}. \tag{7}$$

The rotary and softmax corrections are separately identifiable; both vanish in the appropriate small-displacement limit.

Disjoint rotary bands have additive score increments, $d^{b\cup c} = d^b + d^c$, because their generators have disjoint block support. Their attention responses need not add. With $F(s) = \sum_j p_j(s)v_j$, their interaction is $F(s + d^b + d^c) - F(s + d^b) - F(s + d^c) + F(s)$. It is the double integral of D^2F over the two score directions, with

$$D^2F(s)[a, b] = \mathbb{E}_p[(a - \mathbb{E}_p a)(b - \mathbb{E}_p b)(v - F(s))].$$

The coupling is therefore attributable to softmax normalisation even when the rotary score components are independent (Appendix D).

3 Joint responses, error certificates, and effective weights

3.1 The interaction omitted by separate attribution

The positional calculation specifies one important class of d_j. Theorem 1 also permits arbitrary prescribed key and value edits. Decomposing its second sum gives

$$\begin{aligned} K_f &= \sum_j (p'_j - p_j)(v_j - y), \qquad V_0 = \sum_j p_j \varepsilon_j, \\ C_{KV} &= \sum_j (p'_j - p_j)\varepsilon_j, \qquad \Delta y = K_f + V_0 + C_{KV}. \end{aligned} \tag{8}$$

Adding separate finite-key and baseline-weighted value responses omits exactly C_{KV}. This sum is AtP* (Kramár et al., 2024) applied to the edited set: its QK fix recomputes softmax at fixed values, the fixed-value case of Theorem 1, and its values enter linearly. The exact response is therefore AtP* plus C_{KV}. The latter measures how a value edit's effect changes when attention is reallocated; its four-cell form is $y(p', v + \varepsilon) - y(p', v) - y(p, v + \varepsilon) + y(p, v)$. Along a joint edit path of strength α, its leading term is

$$C_{KV}(\alpha) = \alpha^2 \sum_j p_j(d_j - \mathbb{E}_p d)\varepsilon_j + O(\alpha^3).$$

The quadratic comparator in our experiments keeps this leading interaction and the exact key response; the finite formula retains the whole interaction.

3.2 A certificate for the local approximation

Let $L_s = \sum_j p_j d_j(v_j - y)$ and $w_j = p_j(e^{t_j} - 1 - t_j)$. Exponential convexity gives $w_j \geq 0$, and normalisation yields

$$\sum_j w_j = -\sum_j p_j t_j = \mathrm{KL}(p\|p'), \qquad K_f - L_s = \sum_j w_j(v_j - y). \tag{9}$$

For a fixed gradient g after output projection W_o, put $a_j = \langle g, (v_j - y)W_o\rangle$ and $b_j = \langle g, \varepsilon_j W_o\rangle$. Then

$$\begin{aligned} |\langle g, [\Delta y - (L_s + V_0)]W_o\rangle| \leq\ & \mathrm{KL}(p\|p') \max_j |a_j| \\ & + \mathrm{TV}(p, p')(\max_j b_j - \min_j b_j). \end{aligned} \tag{10}$$

The first term bounds the softmax remainder; the second bounds the joint interaction. A positional-Jacobian certificate additionally accounts for the rotary remainder in (7). When two AtP scores differ by more than the sum of their bounds, their exact local ordering is certified. The proof and sparse formulas appear in Appendix D.

3.3 Every head computes an exact gradient step

Return to affine projections of an input x_i, and augment it as $u_i = (x_i^\top, 1)^\top$. RoPE gives bilinear scores $s_{ij} = u_i^\top a_{ij}$, where $a_{ij} = \widehat{W}_q R(p_i)^\top \widetilde{k}_j^\top/\tau$. Let μ_i be the uniform mean of the attended values,

the readout of uniform attention. Applying $e^s - 1 = s\varphi_1(s)$ between uniform and actual scores gives

$$\begin{aligned} c_{ij} &= \varphi_1(s_{ij})/Z_i > 0, \qquad \Delta M_i = \sum_j c_{ij} a_{ij} (v_j - \mu_i), \\ y_i &= \mu_i + u_i^\top \Delta M_i, \qquad \Delta M_i = -\nabla_B \mathcal{E}_i(0), \\ \mathcal{E}_i(B) &= \tfrac{1}{2} \sum_j c_{ij} \left\| a_{ij}^\top B - (v_j - \mu_i) \right\|_2^2 . \end{aligned} \tag{11}$$

Every head therefore takes one exact gradient step: for any weights and query, $\mathcal{E}_i$ regresses the centred values on the features with positive weights set by the query, and one unit step from $B = 0$ gives the effective matrix ΔM_i, whose prediction at u_i the head adds to the uniform readout. Its rank is at most d; for small scores it is linear-attention gradient descent (Appendix B).

The positive curvature has an operational consequence. Subtract one reference key feature from every score feature, which leaves softmax unchanged. Let Γ_i be the quadratic's feature curvature restricted to the m editable input coordinates, and let H_{all} stack the corresponding relative-score features. Positivity gives

$$\ker \Gamma_i = \ker H_{\mathrm{all}} = \{h : p(x_i + h) = p(x_i)\}. \tag{12}$$

It holds for finite query edits at a fixed bank and position, with the same kernel for every query and reference key, and underlies the feasible control of Section 4 (Appendix C.4). Because the weights depend on the query, reusing an anchor matrix ΔM_a omits exactly $u_i^\top (\Delta M_i - \Delta M_a)$ (Appendix C).

3.4 State-dependent weights beyond fixed affine prediction

An exact weighted readout need not be a fixed linear predictor: its weights can depend on an observed state. The following construction isolates the resulting prediction advantage on two lagged observations. The comparison holds the available inputs fixed.

Theorem 2 (State-dependent readout beyond fixed affine prediction)**.** *Let X_1, X_2 be independent, centred scalar observations with unit variance. Let Z independently take values $-1, +1$ with equal probability, and let ε be independent, centred noise with variance σ_ε^2. Define*

$$Y = \frac{X_1 + X_2}{2} + \frac{Z(X_1 - X_2)}{2} + \varepsilon, \qquad \mathcal{R}(f) = \mathbb{E}[(Y - f)^2].$$

For the affine class $\mathcal{L} = \{b_0 + b_1 X_1 + b_2 X_2 + b_3 Z : b \in \mathbb{R}^4\}$,

$$\inf_{L \in \mathcal{L}} \mathcal{R}(L) = \sigma_\varepsilon^2 + \tfrac{1}{2}. \tag{13}$$

Suppose two fixed lags admit distinct relative RoPE rotations of a shared key. A single head attending to those two entries, with values X_1, X_2 and an affine query in Z, can realise, for every $\kappa > 0$,

$$\begin{aligned} T_\kappa &= \mathrm{sigmoid}(\kappa Z) X_1 + [1 - \mathrm{sigmoid}(\kappa Z)] X_2, \\ \mathcal{R}(T_\kappa) &= \sigma_\varepsilon^2 + \frac{2}{(1 + e^\kappa)^2} \longrightarrow \sigma_\varepsilon^2. \end{aligned} \tag{14}$$

Its readout has the exact decomposition

$$T_\kappa = \frac{X_1 + X_2}{2} + \frac{\tanh(\kappa/2)}{2} Z(X_1 - X_2). \tag{15}$$

Every $\kappa > 0$ improves on the affine error floor (Appendix N.4 gives the proof). The improvement is the state–observation interaction in (15); its weights and finite readout responses remain explicit. Supplying ZX_1, ZX_2 to a linear readout removes the obstruction. The theorem therefore separates state-dependent weighting from fixed affine prediction on the raw inputs.

4 Constructing and ranking interventions

The calculus supports two questions: which query edit realises a requested attention relationship, and which candidate edit best serves a specified objective? Both use the same captured head and exact local response.

Construct a requested relationship. For token pairs (j_r, k_r), request log attention ratios ℓ_r^*. Apply any prescribed rotary shift first, obtaining score features a'_j. Let H have rows $(\bar{a}'_{j_r} - \bar{a}'_{k_r})^\top$, where $\bar{a}'_j$ contains the first m coordinates of a'_j, and put $b_r = \ell_r^* - u^\top(a'_{j_r} - a'_{k_r})$. An edit h of the affine query input then satisfies the requested ratios exactly when $Hh = b$. If this system is feasible, its unique minimum-Euclidean-norm edit is

$$h_* = H^\dagger b, \qquad HH^\dagger b = b. \tag{16}$$

Thus one can set $A{:}B = 4{:}1$, preserve $C{:}D$, and compensate for a specified positional shift in one solve. Feature differences make the construction gauge invariant; with all key pairs included, $\ker H$ is exactly the set of query directions preserving the attention distribution. Appendix C.5 gives the proof and infeasibility criterion.

Evaluate the local response. For score edits supported on S and value edits on T, let $r_j = p_j d_j \varphi_1(d_j)$ on S. The exact response is

$$\Delta y = \frac{\sum_{j\in S} r_j(v_j - y) + \sum_{j\in T} p_j\varepsilon_j + \sum_{j\in S\cap T} r_j\varepsilon_j}{1 + \sum_{j\in S} r_j}. \tag{17}$$

The baseline readout, probabilities, and candidate-group complement statistics are shared. Each further evaluation uses only the edited entries. Stable evaluation uses log probabilities, `expm1`, and positive complement masses. Preparation and reuse are timed separately.

Rank candidates and execute selected edits. Choose a scalar answer margin m and a candidate family. One baseline backward pass supplies $g_\ell = \nabla_{o_\ell} m$ at the projected write $o_\ell = \sum_h y_{\ell h} W_{o,\ell h}$. Score each candidate c by

$$\widehat{\Delta m}_c = \left\langle g_\ell, \sum_h \Delta y_{c,h} W_{o,\ell h} \right\rangle, \qquad \Delta m_c = m_c^{\text{executed}} - m^{\text{baseline}}. \tag{18}$$

Sorting the predicted improvements gives an execution order without an additional model call per candidate. Native execution measures the actual margin change; an external task check determines whether a proposed answer repair succeeds. The approximation in this prediction is propagation through the remaining network with the frozen gradient, the infinitesimal limit of the exact secant factorisation in Appendix D.8.

Match the decision to its objective. Deleting a proper token group G, of mass $\alpha_G < 1$ and weighted value w_G, gives $\Delta y_{-G} = (\alpha_G y - w_G)/(1 - \alpha_G)$. Its projected write norm can be constrained by a specified distortion tolerance, giving a checkable current-query eviction criterion. For precision restoration with an original cache available, let e be the current projected-write error and $\Delta_j o$ the exact change from restoring entry j. The benefit $B_j = \|e\|_2^2 - \|e + \Delta_j o\|_2^2$ ranks restorations by their exact reduction in local squared error. Re-scoring after each restoration defines a greedy procedure for that objective; answer repair instead uses the margin score and executed task check.

5 Experiments

We test the predictions against executed positional and joint-cache interventions, then evaluate ranking, precision restoration, and local evaluation cost.

5.1 Finite positional prediction and its frequency dependence

We evaluate Qwen2.5-1.5B-Instruct and SmolLM2-1.7B-Instruct (Allal et al., 2025) using FP32 eager attention. Each model is evaluated on 256 held-out prompt sets per task: corrupted category mapping (Corrupt), ordered register assignment (Ordered), and SST-2 sentiment. Calibration is performed separately for each model, with a shared policy across tasks fixed before testing. Each set has five variants, two shifts $\{16, 128\}$, and four band choices. The retained layers are $\{0, 14, 21\}$ for Qwen

Table 1: **Prediction of executed positional effects.** Downstream margin MAE on 256 held-out prompt sets per task and model. Intervals are paired 95% prompt-bootstrap intervals for the reduction relative to zero. Within each model, all predictors share the baseline gradient.

Task	Finite	Jacobian	Zero	Reduction vs. zero
Qwen2.5-1.5B-Instruct				
Corrupt	0.02808	0.16028	0.05724	50.9% [48.2, 53.7]
Ordered	0.05786	0.21897	0.09073	36.2% [35.0, 37.5]
SST-2	0.03585	0.15426	0.06681	46.3% [43.1, 49.9]
SmolLM2-1.7B-Instruct				
Corrupt	0.003230	0.186294	0.028152	88.5% [88.1, 89.0]
Ordered	0.007356	0.264973	0.054455	86.5% [86.3, 86.7]
SST-2	0.002376	0.135124	0.029654	92.0% [91.6, 92.3]

and $\{12, 18\}$ for SmolLM2, giving 92,160 and 61,440 executed positional edits, respectively. Keys are rotated at one readout; query, values, and mask are fixed there. Errors are averaged within prompt sets, and 5,000 paired prompt-level bootstrap resamples retain all interventions in their original clusters. Appendix G supplies the full protocol.

Finite responses reduce Jacobian MAE by 73.6–82.5% on Qwen and 97.2–98.3% on SmolLM2 (Table 1), with sign accuracies of 95.36–96.52% and 98.11–98.60%, respectively (Appendices G.2 and G.3). Finite prediction also improves on zero in every model–task setting; the zero predictor's MAE measures the magnitude of the executed effect.

The Qwen angular strata show where the correction matters. At shift 16, maximum fast-, middle-, and slow-band angles are 16, 0.1386, and 0.001489 radians. In the slow band, both predictors reach MAEs of $4\text{–}6 \times 10^{-6}$ at native precision. In the middle band, the Jacobian already removes 94.0–94.8% of zero-predictor error; finite MAEs of $4.0\text{–}6.4 \times 10^{-5}$ further reduce the small remaining error. Large fast-band shifts are deliberately outside the tangent regime. At middle-band shift 128, finite MAE is 0.002112–0.003543 against Jacobian MAE 0.016428–0.027925, for effects of mean magnitude 0.027545–0.038921. Full strata, including SmolLM2's on its own rotary frequencies, are retained in Appendix L. The angular strata identify where finite corrections have a material absolute effect.

The Qwen experiments also measure the predicted structural distinctions. Fast–middle interaction contributes 3.33–9.20% of joint local-response norm at shift 16 and 16.20–26.94% at shift 128. Under actual prompt reordering, direct phase accounts for essentially all of the layer-0 response, whereas changed contextual features have signed projections of 0.935–0.976 onto the total response at layers 14 and 21. Thus frozen-feature phase edits and prompt reorderings answer different experimental questions.

5.2 The joint interaction improves downstream prediction

The joint-cache evaluation contains six Qwen settings and four second-model validation settings. Qwen2.5-0.5B-Instruct is evaluated at layer 18 and Qwen2.5-1.5B-Instruct at layers 14 and 21; SmolLM2-1.7B-Instruct is evaluated at layers 12 and 18. Every checkpoint–layer combination is tested on retrieval and SST-2 using 32 prompt pairs and eight spans, with no filtering by accuracy or outcome. The Qwen sweep contains 5,120 interventions, including two additional joint-edit strengths at 1.5B retrieval layer 21. The SmolLM2 validation adds 3,584 interventions, including the same two fractional strengths at retrieval layer 18, for 8,704 in total.

Retaining C_{KV} lowers error by 83.8–99.6% against AtP* and by 75.9–99.0% against the quadratic correction across the ten settings, and by 92.7–99.6% against AtP in the six Qwen settings, where AtP's own MAE is 38–114% of the mean executed effect, the zero predictor's MAE (Table K.1). All 26 paired 95% intervals against these comparators favour exact joint. At Qwen2.5-1.5B layer 21, AtP* is less accurate than predicting no change on both tasks, whereas exact joint reduces the zero predictor's MAE by 92.1% and 95.8%. Exact joint differs from AtP* by exactly C_{KV}, whose contraction correlates 0.99447–0.99999 with the independently executed downstream four-cell interaction, measured after executing the remaining network. The SmolLM2 extension therefore reproduces the directional result in a second model family.

Table 2: **Joint-edit prediction across ten settings and two model families.** Downstream margin MAE over 32 prompt pairs and eight joint edits each. AtP* applies its QK fix to the whole span and adds baseline-weighted values. Quadratic adds the leading mixed term; exact adds C_{KV}. Complete Qwen controls and paired intervals are in Tables K.1 and K.2; SmolLM2 intervals are in Table K.4.

Checkpoint	Task	Layer	AtP*	Quadratic	Exact
Qwen2.5-0.5B	Retrieval	18	0.001042	2.98×10^{-4}	4.37×10^{-6}
Qwen2.5-0.5B	SST-2	18	0.003099	5.99×10^{-4}	2.64×10^{-5}
Qwen2.5-1.5B	Retrieval	14	0.003342	0.001525	8.22×10^{-5}
Qwen2.5-1.5B	Retrieval	21	0.107864	0.087683	0.007844
Qwen2.5-1.5B	SST-2	14	0.003194	0.001418	3.42×10^{-4}
Qwen2.5-1.5B	SST-2	21	0.109616	0.080700	0.004510
SmolLM2-1.7B	Retrieval	12	6.74×10^{-4}	4.67×10^{-4}	1.48×10^{-5}
SmolLM2-1.7B	Retrieval	18	0.032226	0.024695	2.41×10^{-4}
SmolLM2-1.7B	SST-2	12	0.003620	0.002736	5.87×10^{-4}
SmolLM2-1.7B	SST-2	18	0.095748	0.044898	0.007711

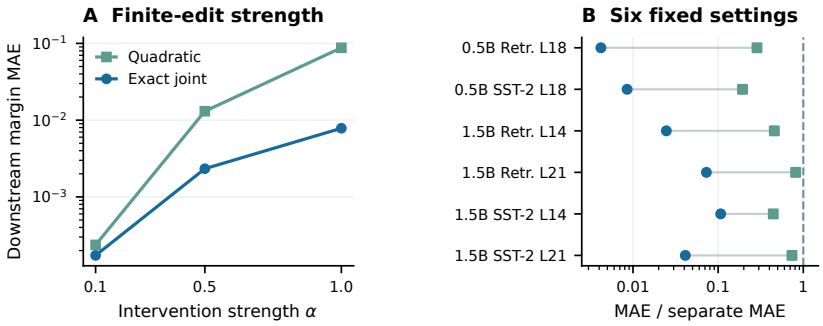

Figure 2: **The joint interaction improves executed-effect prediction.** Left: downstream MAE by edit strength on 1.5B retrieval, layer 21. Right: MAE relative to separate attribution (AtP*) in six Qwen settings. Table 2 gives absolute values and Table K.2 gives paired intervals.

At strengths $\alpha = 0.1, 0.5, 1$, exact-joint MAEs are 0.000173, 0.002330, and 0.007844; quadratic MAEs are 0.000237, 0.013071, and 0.087683. At the smallest strength, both recover the two most influential spans perfectly. The gap grows with edit strength, while the exact predictor's own error also grows (Figure 2). Appendix F gives the complete execution protocol and additional plots.

5.3 Ranking and Repair Decisions

Deletion-effect ranking attains Spearman correlations of 0.950–0.989 against executed effects, above the Jacobian, attention-mass, and value-norm controls in every condition. The matched single-entry precision-restoration comparison reduces mean relative RMSE by 4.86%, 8.13%, and 6.67% in three task/precision conditions, with paired difference intervals below zero; across all six, the point estimate improves in five and ties in one. Four-entry greedy restoration improves 19.8–54.9% against fixed second-order selection. This evaluates the complete greedy procedure, including re-scoring after each restoration (Appendix G).

The accompanying instrument also records a selected synthetic API repair on Qwen2.5-0.5B-Instruct. Ordinary generation passes 0/6 fixed checks, source emphasis passes 1/6, and an installed no-op reproduces the 0/6 failure. A ranked attention edit produces a program that passes 6/6. This example illustrates the score–execute–check workflow (Appendix H).

5.4 Preparation, reuse, and numerical agreement

Prepared sparse scoring takes 0.336 ms versus 0.925 ms on the native A100 case. Including preparation, it is slower for one strength but 18.0–21.8% faster across seven workloads when three strengths reuse the same preparation (Appendix F). Native Qwen and SmolLM2 reconstructions, sparse–dense comparisons, and 1,586,464 positional checks verify implementation agreement (Appendix E).

5.5 Controlled numeric readouts

A trained RoPE head reads sixteen lagged scalars x_j with query $q(z) = q_0 + zq_1$ and predicts

$$\widehat{y} = b + \beta \sum_j p_j(z) x_j = b + \sum_j w_j(z) x_j, \qquad w_j(z) = \beta p_j(z). \tag{19}$$

With no nonlinear block after the head, local changes are exact forecast changes. Across five autoregressive runs, 11,807 interventions yield 47,066 passing checks of weights, masking, rotary identities, response decompositions, and scenario reuse; the largest residual is 8.64×10^{-13}. Including preparation, evaluation is 6.68 times faster at 1,024 entries and 256 scenarios.

We also train the two-lag head from Theorem 2 on ten independent switching-task replicates. Its mean test MSE is 0.010193 versus 0.516066 for raw-input OLS, a paired reduction of 0.505873 with 95% interval $[0.498771, 0.512841]$. On a linear control the reduction is -5.36×10^{-6} with interval $[-2.09 \times 10^{-5}, 8.15 \times 10^{-6}]$. An OLS comparator supplied the interaction features ZX_1, ZX_2 also reaches the noise floor, isolating state-dependent weighting. Appendix N gives all protocols, aggregate and per-seed results, and 8,380 additional numerical checks.

6 Related work

RoPE supplies the relative-position rotation (Su et al., 2021), and φ_1 is classical in exponential integration (Hochbruck & Ostermann, 2010). Together they give finite attention responses, an error decomposition, and intervention rules; on attention weights, φ_1 yields the Sato–Vartia logarithmic means (Sato, 1976; Vartia, 1976). Every softmax head is an exact gradient step on our quadratic, with the centred linear-attention step of von Oswald et al. (2023) as its small-score limit; this complements functional gradient descent (Cheng et al., 2024), fast weights (Schlag et al., 2021), implicit updates (Dai et al., 2023), Hopfield correspondences (Ramsauer et al., 2021), and kernel regression (Nadaraya, 1964; Watson, 1964); Appendix I compares an independent regression learner.

Attribution patching supplies the shared-gradient scoring framework (Syed et al., 2024). AtP* recomputes the softmax for a patched query or key at fixed values and patches values linearly (Kramár et al., 2024). Our calculus contains this QK fix as its fixed-value case, derives the score increments of finite positional edits from the RoPE generator, and retains the key–value interaction AtP* omits, reducing its error by 83.8–99.6%. EAP-IG approximates a finite gradient by path integration (Hanna et al., 2024); endpoint secants give an exact one in closed form (Appendix D.8). Attribution graphs freeze attention, keeping OV but not QK circuits (Elhage et al., 2021; Ameisen et al., 2025), and so omit K_f and C_{KV}. PASTA steers attention toward specified spans (Zhang et al., 2024); our emphasis is finite responses, feasible controls, scoring, and execution. The distortion criteria also support attention-based cache selection such as SnapKV (Li et al., 2024) and reuse of cached chunks at new positions (Yao et al., 2025; Hu et al., 2025). We specify endpoints, controls, metrics, and statistical units following Zhang & Nanda (2024).

7 Discussion

Taken between two states, the compact identity exposes the attribution interaction AtP* omits, the exact gradient step inside every softmax head, a geometry for attention edits, local criteria for cache and approximation decisions, and an exact account of downstream propagation, without a learned response or small-edit expansion. In every evaluated setting, retaining finite effects improves downstream prediction, while frequency and strength sweeps identify when the correction matters. The result is a reviewable workflow: specify an edit, rank its local response, and check selected candidates by execution.

REPRODUCIBILITY STATEMENT

The appendices preserve the complete derivations, model revisions, prompt construction, calibration policy, numerical tolerances, and prompt-paired evaluation procedures from the full study. Appendix K gives the fixed joint-KV sweep and full tables; Appendix M describes the standalone runner, native preflights, and replay captures. Appendix G.3 documents the separately calibrated SmolLM2 positional evaluation, including the pinned checkpoint, selected layers, numerical checks, and prompt-cluster uncertainty estimates. The source includes a figure-generation script and the reported summary values used in the empirical plots; the tables retain the paired intervals. Experimental code, saved result archives, and the instrument are separate research artifacts supporting this submission.

AI USE STATEMENT

Generative AI tools were used only to edit the writing of this manuscript.

REFERENCES

Reduan Achtibat, Sayed Mohammad Vakilzadeh Hatefi, Maximilian Dreyer, Aakriti Jain, Thomas Wiegand, Sebastian Lapuschkin, and Wojciech Samek. AttnLRP: Attention-aware layer-wise relevance propagation for transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 135–168, 2024.

Ameen Ali, Thomas Schnake, Oliver Eberle, Grégoire Montavon, Klaus-Robert Müller, and Lior Wolf. XAI for transformers: Better explanations through conservative propagation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 435–451, 2022.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM2: When Smol Goes Big—Data-Centric Training of a Small Language Model, 2025. URL https://arxiv.org/abs/2502.02737.

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/methods.html.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations*, 2018.

T. L. Bennet. The theory of measurement of changes in cost of living. *Journal of the Royal Statistical Society*, 83(3):455–462, 1920.

Xiang Cheng, Yuxin Chen, and Suvrit Sra. Transformers implement functional gradient descent to learn non-linear functions in context. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8002–8037, 2024.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 276–286. Association for Computational Linguistics, 2019. doi: 10.18653/v1/W19-4828.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4005–4019. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.247. URL https://aclanthology.org/2023.findings-acl.247/.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

James R. Golden. Equivalent linear mappings of large language models. *Transactions on Machine Learning Research*, 2025. arXiv:2505.24293.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. In *First Conference on Language Modeling*, 2024.

Marlis Hochbruck and Alexander Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010. doi: 10.1017/S0962492910000048.

Junhao Hu, Wenrui Huang, Weidong Wang, Haoyi Wang, Tiancheng Hu, Zhang Qin, Hao Feng, Xusheng Chen, Yizhou Shan, and Tao Xie. EPIC: Efficient position-independent caching for serving large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24391–24402, 2025.

János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. AtP*: An efficient and scalable method for localizing LLM behaviour to components. arXiv:2403.00745, 2024. URL https://arxiv.org/abs/2403.00745.

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. arXiv:2404.14469, 2024. URL https://arxiv.org/abs/2404.14469.

Jan Maas. Gradient flows of the entropy for finite Markov chains. *Journal of Functional Analysis*, 261(8):2250–2292, 2011. doi: 10.1016/j.jfa.2011.06.009.

E. A. Nadaraya. On estimating regression. *Theory of Probability and Its Applications*, 9(1):141–142, 1964. doi: 10.1137/1109020.

Qwen Team. Qwen2.5-1.5B-Instruct: Model card, 2024a. URL https://huggingface.co/Qwen/Qwen2.5-1.5B-Instruct.

Qwen Team. Qwen2.5-0.5B: Model card, 2024b. URL https://huggingface.co/Qwen/Qwen2.5-0.5B.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2008.02217.

Kazuo Sato. The ideal log-change index number. *The Review of Economics and Statistics*, 58(2): 223–228, 1976.

Hermann Schichl and Arnold Neumaier. Interval analysis on directed acyclic graphs for global optimization. *Journal of Global Optimization*, 33(4):541–562, 2005. doi: 10.1007/s10898-005-0937-x.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9355–9366. PMLR, 2021. URL https://proceedings.mlr.press/v139/schlag21a.html.

Jacob Schreiber. tangermeme, version 1.5.0. Software release, 25 September 2026, 2026. URL https://github.com/jmschrei/tangermeme/releases/tag/v1.5.0.

Lingfeng Shen, Aayush Mishra, and Daniel Khashabi. Position: Do pretrained transformers learn in-context by gradient descent? In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3145–3153, 2017.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642. Association for Computational Linguistics, 2013. URL https://aclanthology.org/D13-1170/.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding, 2021. URL https://arxiv.org/abs/2104.09864. arXiv:2104.09864.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3319–3328, 2017.

Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. In *Proceedings of the 7th BlackboxNLP Workshop*, pp. 407–416. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.blackboxnlp-1.25. URL https://aclanthology.org/2024.blackboxnlp-1.25/.

Yrjö O. Vartia. Ideal log-change index numbers. *Scandinavian Journal of Statistics*, 3:121–126, 1976.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, Joao Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 35151–35174. PMLR, 2023. URL https://proceedings.mlr.press/v202/von-oswald23a.html.

Geoffrey S. Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, 26(4):359–372, 1964. URL https://www.jstor.org/stable/25049340.

Jiayi Yao, Hanchen Li, Yuhan Liu, Siddhant Ray, Yihua Cheng, Qizheng Zhang, Kuntai Du, Shan Lu, and Junchen Jiang. CacheBlend: Fast large language model serving for RAG with cached knowledge fusion. In *Proceedings of the Twentieth European Conference on Computer Systems (EuroSys '25)*. ACM, 2025. doi: 10.1145/3689031.3696098.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.16042.

Luyang Zhang and Jialu Wang. When attribution patching lies: Diagnosis and a second-order correction. *arXiv preprint arXiv:2606.09899*, 2026.

Qingru Zhang, Chandan Singh, Liyuan Liu, Xiaodong Liu, Bin Yu, Jianfeng Gao, and Tuo Zhao. Tell your model where to attend: Post-hoc attention steering for LLMs. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2311.02262.

A RoPE DERIVATIVES AND FINITE DIFFERENCES

For fixed content z_0, let $z(p) = R(p)z_0$ be a rotated column feature. The generator in (2) gives

$$\partial_p^n z(p) = A^n z(p), \qquad n \geq 1. \tag{A.1}$$

The matrix extension of (B.5) is defined by the convergent power series $\rho(B) = \sum_{n\geq 0} B^n/(n+1)!$. Consequently

$$\boxed{e^{\delta A} - I = \delta A \rho(\delta A), \qquad z(p+\delta) - z(p) = \delta A \rho(\delta A) z(p).} \tag{A.2}$$

The power-series definition also covers zero-frequency blocks. These identities expose RoPE as a bank of position-response operators on frozen features: A gives the infinitesimal response and $A\rho(\delta A)$ gives the exact finite-difference quotient. The same entire function converts the scalar softmax increment $e^s - 1$ into $s\rho(s)$ and the matrix rotary increment $e^{\delta A} - I$ into $\delta A \rho(\delta A)$. In (B.7), rotary position structure enters through $R(p_j - p_i)$ and its scores, while $\rho(s_{ij})$ retains their full exponential response. The derivatives in (A.1) are with respect to position at fixed content.

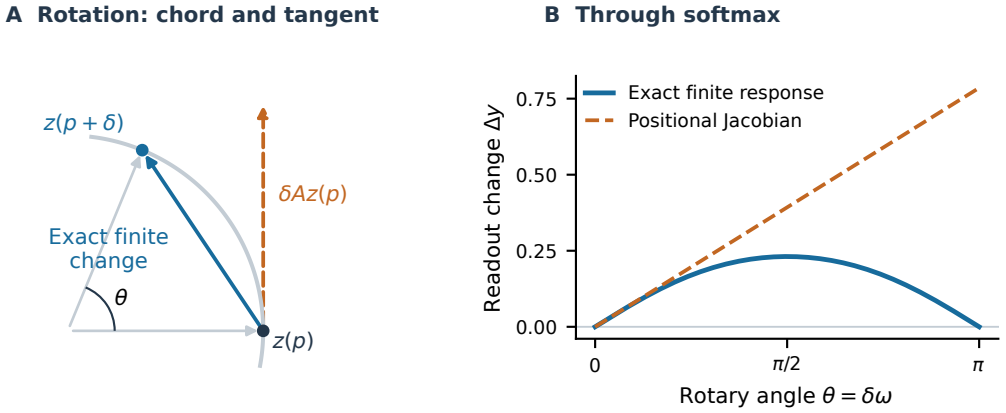

Figure A.1: **The derivative determines the tangent; its integral gives the finite response.** Left: a single rotary plane, with the exact chord and the tangent step at the same displacement. Right: an analytic two-key example with scores $(\sin\theta, 0)$ and scalar values $(1, 0)$. The exact readout change is $\mathrm{sigmoid}(\sin\theta) - 1/2$; the positional Jacobian gives $\theta/4$. This illustration separates the mathematical issue from downstream model behaviour; it is not an empirical model measurement.

B EXACT EFFECTIVE WEIGHTS WITH SOFTMAX INTACT

B.1 RoPE, AFFINE PROJECTIONS, AND THE ATTENDED BANK

Let $x_i \in \mathbb{R}^m$ be a column input, with query/key width d and value width r. Use row-vector projections

$$q_i = x_i^\top W_q + b_q, \qquad k_j = x_j^\top W_k + b_k, \qquad v_j = x_j^\top W_v + b_v, \tag{B.1}$$

where $W_q, W_k \in \mathbb{R}^{m\times d}$, $W_v \in \mathbb{R}^{m\times r}$, and biases are rows. For standard RoPE with fixed frequencies (Su et al., 2021), set $\tau = \sqrt{d}$ and

$$R(p) = e^{pA}, \qquad A = \mathrm{diag}_{\ell=1}^{d/2} \begin{pmatrix} 0 & -\omega_\ell \\ \omega_\ell & 0 \end{pmatrix}, \qquad R(p_i)^\top R(p_j) = R(p_j - p_i). \tag{B.2}$$

The rotated rows are $\widetilde{q}_i = q_i R(p_i)^\top$ and $\widetilde{k}_j = k_j R(p_j)^\top$. Augment the query input and projection by

$$u_i = \begin{pmatrix} x_i \\ 1 \end{pmatrix}, \qquad \widehat{W}_q = \begin{pmatrix} W_q \\ b_q \end{pmatrix}, \qquad L_i = \frac{\widehat{W}_q R(p_i)^\top}{\tau}, \qquad a_{ij} = L_i \widetilde{k}_j^\top. \tag{B.3}$$

Then the raw score is exactly $s_{ij} = \widetilde{q}_i \widetilde{k}_j^\top / \tau = u_i^\top a_{ij}$. In the bias-free case this is $s_{ij} = x_i^\top W_q R(p_j - p_i) W_k^\top x_j / \tau$. Let $\mathcal{I}_i$ be a finite nonempty set of permitted keys; causal masking selects this set. For deterministic attention, define

$$N_i = |\mathcal{I}_i|, \qquad Z_i = \sum_{j \in \mathcal{I}_i} e^{s_{ij}}, \qquad \mu_i = \frac{1}{N_i} \sum_{j \in \mathcal{I}_i} v_j, \qquad y_i = \frac{1}{Z_i} \sum_{j \in \mathcal{I}_i} e^{s_{ij}} v_j. \tag{B.4}$$

Here μ_i is the uniform arithmetic mean of the permitted values.

B.2 The exponential divided difference

The entire function

$$\rho(s) = \varphi_1(s) = \exp[0, s] = \int_0^1 e^{ts} \, \mathrm{d}t = \sum_{n=0}^{\infty} \frac{s^n}{(n+1)!} = \begin{cases} (e^s - 1)/s, & s \neq 0, \\ 1, & s = 0 \end{cases} \tag{B.5}$$

is the first divided difference of the exponential at 0 and s, with its continuous value at coincident arguments. It is the standard φ_1 of exponential integration (Hochbruck & Ostermann, 2010); $\rho(s) > 0$ for real s and $e^s - 1 = s\rho(s)$.

Theorem B.1 (Exact effective-weight representation)**.** *For the head in* (B.1)–(B.4)*, define*

$$\boxed{c_{ij} = \frac{\rho(s_{ij})}{Z_i}, \qquad \Delta M_i = \sum_{j \in \mathcal{I}_i} c_{ij} a_{ij} (v_j - \mu_i), \qquad y_i = \mu_i + u_i^\top \Delta M_i.} \tag{B.6}$$

The last equality holds exactly for every query and all finite projection weights and inputs.

Proof. The centred values sum to zero. Therefore

$$\begin{aligned} y_i - \mu_i &= Z_i^{-1} \sum_j e^{s_{ij}} (v_j - \mu_i) = Z_i^{-1} \sum_j (e^{s_{ij}} - 1)(v_j - \mu_i) \\ &= Z_i^{-1} \sum_j s_{ij} \rho(s_{ij}) (v_j - \mu_i) = u_i^\top \sum_j c_{ij} a_{ij} (v_j - \mu_i). \end{aligned}$$

□

In particular, without projection biases the effective matrix in the original weights is

$$\Delta M_i = \frac{1}{\tau Z_i} \sum_{j \in \mathcal{I}_i} \rho(s_{ij}) \big[W_q R(p_j - p_i) W_k^\top x_j \big] \big[x_j^\top W_v - \mu_i \big]. \tag{B.7}$$

At a fixed bank and query position, write $L = L_i$ and $a_j = a_{ij}$. Then $\Delta M_i = L T_i$, where $T_i = \sum_j c_{ij} \widetilde{k}_j^\top (v_j - \mu) \in \mathbb{R}^{d \times r}$. The cache supplies the outer-product directions; the query selects their positive coefficients. This gives $\operatorname{rank}(\Delta M_i) \leq d$ in this representation.

Corollary B.2 (A query-conditioned unit gradient step)**.** *For each fixed query, define* $B \in \mathbb{R}^{(m+1) \times r}$ *and*

$$\mathcal{E}_i(B) = \frac{1}{2} \sum_{j \in \mathcal{I}_i} c_{ij} \left\| a_{ij}^\top B - (v_j - \mu_i) \right\|_2^2. \quad \textit{Then} \quad 0 - \nabla_B \mathcal{E}_i(0) = \Delta M_i. \tag{B.8}$$

Proof. With the query, features, centred values, and coefficients held fixed, $\nabla_B \mathcal{E}_i(B) = \sum_j c_{ij} a_{ij} (a_{ij}^\top B - (v_j - \mu_i))$; evaluation at $B = 0$ gives the result. □

A unit step from zero on $\mathcal{E}_i$ gives the effective matrix and its exact readout $y_i = \mu_i + u_i^\top \Delta M_i$. The regression features are a_{ij} and the responses are the centred cached values $v_j - \mu_i$; every coefficient is computed directly from the query and its bank through (B.6). The step therefore identifies a matrix in input coordinates and the tokenwise outer products that form it. At a fixed bank and position, changing the query reweights these same directions. Section C makes that change explicit through the score-shift family and the exact residual from reusing an anchor matrix.

The linear-attention limit. Scaling every feature a_{ij} by ϵ scales every score by ϵ. Expanding φ_1 and Z_i gives $c_{ij} = N_i^{-1}(1 + s_{ij}/2 - \bar{s}_i) + O(\epsilon^2)$, where $\bar{s}_i = N_i^{-1}\sum_{k\in\mathcal{I}_i} s_{ik}$, and therefore

$$\Delta M_i = \Delta M_i^{\mathrm{lin}} + \frac{1}{N_i}\sum_{j\in\mathcal{I}_i}\Big(\frac{s_{ij}}{2} - \bar{s}_i\Big)a_{ij}(v_j - \mu_i) + O(\epsilon^3), \qquad \Delta M_i^{\mathrm{lin}} = \frac{1}{N_i}\sum_{j\in\mathcal{I}_i} a_{ij}(v_j - \mu_i). \tag{B.9}$$

Here $\Delta M_i^{\mathrm{lin}} = -\nabla_B \mathcal{E}_i^{\mathrm{lin}}(0)$ for the unweighted loss $\mathcal{E}_i^{\mathrm{lin}}(B) = \frac{1}{2N_i}\sum_j \left\|a_{ij}^\top B - (v_j - \mu_i)\right\|_2^2$, and its readout $\mu_i + N_i^{-1}\sum_j s_{ij}(v_j - \mu_i)$ is linear attention on centred values. This is the centred outer-product form of the linear-attention gradient step; the corresponding regression step is specified by the feature/label encoding, projection choices, and step scale of von Oswald et al. (2023). Softmax attention therefore reduces to this step in the small-score limit, and its first correction reweights each cached example by $1 + s_{ij}/2 - \bar{s}_i$.

C Residual coordinates, gauge, and exact reuse error

C.1 Residual write and score-shift family

In the bias-free case, if the head input is also the residual input and $W_o \in \mathbb{R}^{r\times m}$, then

$$x_i^\top + y_i W_o = x_i^\top(I_m + \Delta M_i W_o) + \mu_i W_o. \tag{C.1}$$

For a pre-normalised head, use $x_i = \mathrm{Norm}(h_i)$ and the exact residual $h_i^\top + (\mu_i + u_i^\top \Delta M_i)W_o + b_o$; projected head writes are summed for multi-head attention. Equation (B.6) is a secant representation of the readout. At a fixed bank and position, its differential is $\mathrm{d}y_i = \mathrm{d}u_i^\top \Delta M_i + u_i^\top \mathrm{d}\Delta M_i$; the second term contains the query derivatives of the coefficients.

Effective matrices depend on a score-shift convention. For any fixed $g \in \mathbb{R}^{m+1}$, put $\sigma_i = u_i^\top g$. Softmax is unchanged by subtracting σ_i from every attended score. Applying Theorem B.1 to the shifted features gives

$$\begin{aligned} a_{ij}^g &= a_{ij} - g, & c_{ij}^g &= \frac{e^{\sigma_i}\rho(s_{ij} - \sigma_i)}{Z_i}, \\ \Delta M_i^g &= \sum_j c_{ij}^g a_{ij}^g (v_j - \mu_i), & y_i &= \mu_i + u_i^\top \Delta M_i^g. \end{aligned} \tag{C.2}$$

The gauge $\sigma_i = \log(Z_i/N_i)$ makes each c_{ij}^g the logarithmic mean of $1/N_i$ and $e^{s_{ij}}/Z_i$ (Appendix D.7). Key shifts give the subfamily $g = L_i b$, of dimension at most d at a fixed position. We call (B.6) the *original-score gauge*. The displayed transformations give a family of exact matrices for the same output map.

C.2 Global affine representability

Proposition C.1 (Characterisation of global affine readouts)**.** *Fix a finite nonempty key/value bank and query position, and let $F : \mathbb{R}^m \to \mathbb{R}^r$ be its softmax head output as the query coordinate ranges over all of $\mathbb{R}^m$. A fixed affine representation $F(x) = b + x^\top M$ exists for every $x \in \mathbb{R}^m$ if and only if F is constant. In that case $M = 0$ and $b = F(0)$.*

Proof. Every output is a convex combination of the fixed values, so $\|F(x)\|_2 \le \max_j \|v_j\|_2$ for all x. If $M \neq 0$, choose z with $z^\top M \neq 0$; then $\left\|b + tz^\top M\right\|_2 \to \infty$ as $|t| \to \infty$. Hence $M = 0$ and F is constant. Conversely, a constant F has the stated affine representation. □

Matrix transfer on a finite query set is quantified by the following exact residual.

C.3 The exact term omitted by freezing

Fix the bank and query position so that a_j and μ are shared, and select an anchor query a. Define $y_i^{\mathrm{fr}} = \mu + u_i^\top \Delta M_a$. Then

$$\boxed{e_i = y_i - y_i^{\mathrm{fr}} = u_i^\top(\Delta M_i - \Delta M_a) = \sum_{j\in\mathcal{I}}(c_{ij} - c_{aj})(u_i^\top a_j)(v_j - \mu).} \tag{C.3}$$

Thus $y_i = \mu + u_i^\top \Delta M_a + e_i$, and $e_i W_o$ is the exact projected correction. Each summand resolves this correction by cached token. In bias-free coordinates, freezing anchor 0 omits exactly $x_i^\top(\Delta M_i - \Delta M_0)$, with

$$\|e_i\|_2 \le \|u_i\|_2 \, \|\Delta M_i - \Delta M_a\|_{\mathrm{op}} . \tag{C.4}$$

The fixed-position condition makes the features shared; changing the position also changes L_i, and changing the bank can change both features and mean.

C.4 Positive curvature and attention-preserving edits

Fix the bank, mask, query position, and affine query projection, and choose a reference key j_0. Suppress the query index and use the score-shift gauge $g = a_{j_0}$ from (C.2). Write

$$\begin{aligned} \widetilde{a}_j &= a_j - a_{j_0}, \qquad \widetilde{s}_j = u^\top \widetilde{a}_j, \qquad \widetilde{c}_j = \frac{\rho(\widetilde{s}_j)}{\sum_k e^{\widetilde{s}_k}} > 0, \\ \mathcal{E}_u^{j_0}(B) &= \frac{1}{2} \sum_j \widetilde{c}_j \left\| \widetilde{a}_j^\top B - (v_j - \mu) \right\|_2^2, \qquad G_u = \sum_j \widetilde{c}_j \widetilde{a}_j \widetilde{a}_j^\top . \end{aligned} \tag{C.5}$$

Theorem B.1 applies in this gauge, giving $y - \mu = u^\top[-\nabla_B \mathcal{E}_u^{j_0}(0)]$. With the coefficients frozen during differentiation, the Hessian acts on a matrix perturbation X as $X \mapsto G_u X$. Let $P = (I_m, 0)^\top$ embed an input edit h as $(h^\top, 0)^\top$, so that the homogeneous bias coordinate is held fixed. Let H_{all} have rows $\widetilde{a}_j^\top P$ over all attended keys, including the zero reference row, and define $\Gamma_u = P^\top G_u P$.

Corollary C.2 (Curvature characterises attention-preserving edits)**.** *Under the fixed-bank assumptions above,*

$$\ker \Gamma_u = \ker H_{\mathrm{all}} = \{h : p(x+h) = p(x)\}. \tag{C.6}$$

This kernel is independent of x and of the chosen reference key.

Proof. Positivity gives

$$h^\top \Gamma_u h = \sum_j \widetilde{c}_j (\widetilde{a}_j^\top P h)^2,$$

which vanishes exactly when every relative score is unchanged. Since $\log(p_j/p_{j_0}) = u^\top \widetilde{a}_j$, these equalities are equivalent to preserving every probability ratio, and hence the positive normalised distribution. The rows of H_{all} span all pairwise feature differences, independently of the reference. The positive weights may change with x, but their null space does not. □

Thus the quadratic exposes an exact distribution-preserving subspace in the editable coordinates. The next construction uses the same feature differences to test feasibility and produce the smallest requested edit. Values can introduce additional output cancellations: preserving the head output is a weaker condition than preserving its attention distribution.

C.5 Gauge-invariant query control

Fix the key/value bank, mask, and affine query projection at one head readout. Apply any prescribed positional key edit first, obtaining $a'_j = ((\bar{a}'_j)^\top, \beta'_j)^\top$. For a query input x and edit h,

$$\log \frac{p_j(x+h)}{p_k(x+h)} = u^\top (a'_j - a'_k) + (\bar{a}'_j - \bar{a}'_k)^\top h. \tag{C.7}$$

All probabilities here refer to the same edited bank. Softmax's common denominator cancels in the ratio, so this equation is affine and exact for arbitrary finite h. Form H and b as in Section 4. The requested log ratios are achievable if and only if $b \in \mathrm{range}(H)$, equivalently $(I - HH^\dagger)b = 0$. In that case every solution is

$$h = H^\dagger b + n, \qquad n \in \ker H.$$

The two terms are orthogonal, giving $\|h\|_2^2 = \|H^\dagger b\|_2^2 + \|n\|_2^2$ and proving the unique minimum-norm claim. With all attended key pairs included, preserving their ratios is equivalent to preserving the entire positive, normalised attention distribution. Thus $\ker H$ gives its invisible query directions and the row space of H gives its controllable query subspace. A common feature shift $a'_j \mapsto a'_j - g$ leaves

both H and b unchanged, so this construction is invariant to the score-shift gauge. In contrast, ΔM_i is a query-dependent secant matrix; its kernel alone need not characterise attention-preserving edits.

Minimum norm refers to the affine head-input coordinate x, with the bank frozen. Realisation through an upstream normalisation or other constrained parameterisation requires the corresponding constrained solve. For inconsistent constraints, $H^\dagger b$ is only a least-squares solution: its nonzero residual must be reported rather than treating the requested ratios as achieved. Numerical implementations use an SVD rank tolerance and verify the achieved ratios against recomputed probabilities.

D Finite changes and cache attribution

At a fixed query, a cache edit changes the effective-matrix readout and may change the uniform value mean. Their combined change is the exact attention increment. We express it in cache coordinates to separate key, value, and interaction effects and obtain the local response used for downstream attribution.

D.1 Exact score and value increments

Fix one query and its attended set, and suppress the query index. Let $p_j = e^{s_j}/Z$, $y = \sum_j p_j v_j$, and consider changed scores and values $s'_j = s_j + \delta_j$, $v'_j = v_j + \varepsilon_j$. Define

$$D = \sum_j p_j e^{\delta_j}, \qquad t_j = \delta_j - \log D, \qquad p'_j = p_j e^{t_j}. \tag{D.1}$$

The same exponential divided difference as in (B.6) gives

$$\boxed{y' - y = \sum_j p_j t_j \rho(t_j)(v_j - y) + \sum_j p'_j \varepsilon_j.} \tag{D.2}$$

Indeed, $p'_j - p_j = p_j(e^{t_j} - 1) = p_j t_j \rho(t_j)$ and $\sum_j (p'_j - p_j) = 0$. Expanding $\sum_j p'_j v'_j - \sum_j p_j v_j$ then proves (D.2). The first term reallocates attention mass and the second changes the attended values; their sum includes their interaction exactly. For numerical evaluation, we compute $t_j = \log p'_j - \log p_j$ using log-softmax. When $t_j \geq 0$, $p'_j - p_j = -p'_j \operatorname{expm1}(-t_j)$; when $t_j < 0$, $p'_j - p_j = p_j \operatorname{expm1}(t_j)$. This evaluates the divided-difference increment without forming an unnormalised exponential sum.

For a positional edit with the augmented query u, values, and attended set fixed, the uniform mean is unchanged. Applying (B.6) before and after the edit gives the direct effective-weight relation

$$\begin{aligned} \Delta y = u^\top(\Delta M' - \Delta M) &= \sum_j (p'_j - p_j)(v_j - \mu) \\ &= \sum_j (p'_j - p_j)(v_j - y) = \sum_j p_j t_j \rho(t_j)(v_j - y). \end{aligned} \tag{D.3}$$

The change of centre uses $\sum_j (p'_j - p_j) = 0$. Equation (D.3) evaluates the effective-weight change directly in cache coordinates.

D.2 Attribution patching and exact joint responses

For the fixed query above, define the score-level linear response

$$L_s = \sum_j p_j \delta_j (v_j - y) = \sum_j p_j t_j (v_j - y), \qquad V_0 = \sum_j p_j \varepsilon_j. \tag{D.4}$$

The equality follows from $\sum_j p_j (v_j - y) = 0$. Contracting $L_s + V_0$ with the baseline downstream gradient, after output projection, gives score/value attribution patching. For finite rotary edits, the positional Jacobian additionally replaces δ_j by δ_j^{lin}.

Set

$$K_f = \sum_j (p'_j - p_j)(v_j - y), \qquad C_{KV} = \sum_j (p'_j - p_j)\varepsilon_j. \tag{D.5}$$

The fixed-value QK fix of AtP* (Kramár et al., 2024) uses K_f before the gradient contraction. Indeed, its probability-gradient contraction can be written

$$\sum_j (p'_j - p_j)\langle g, v_j W_o\rangle = \sum_j (p'_j - p_j)\langle g, (v_j - y)W_o\rangle. \tag{D.6}$$

This equality is for the sum; the right-hand side defines the centred token allocation used here. Equation (D.2) now reads

$$\boxed{y' - y = K_f + V_0 + C_{KV}.} \tag{D.7}$$

Thus AtP*, which adds a fixed-value key correction and a baseline-weighted value correction, omits exactly C_{KV}. In a four-cell intervention on probabilities and values, this term is

$$C_{KV} = y(p', v + \varepsilon) - y(p', v) - y(p, v + \varepsilon) + y(p, v). \tag{D.8}$$

It measures the change in a value edit's effect when attention is reallocated. For joint cache edits in the original effective-weight coordinates, the uniform mean also changes, giving

$$y' - y = (\mu' - \mu) + u^\top(\Delta M' - \Delta M) = K_f + V_0 + C_{KV}. \tag{D.9}$$

The query is fixed in this equality; the key and value bank changes. Equation (D.9) connects changes in the input-coordinate map to separate key, value, and interaction contributions evaluated in cache coordinates.

The exact score-linearisation remainder is

$$R_s = K_f - L_s = \sum_j p_j t_j \big(\rho(t_j) - 1\big)(v_j - y). \tag{D.10}$$

Consequently, the joint response decomposes into $L_s + V_0$, the softmax remainder R_s, and the interaction C_{KV}. A generator-linearised positional comparator has the additional remainder $\sum_j p_j(\delta_j - \delta_j^{\mathrm{lin}})(v_j - y)$.

Proposition D.1 (KL mass of the local remainder)**.** *Let p, p' have positive entries on the fixed attended set and define*

$$w_j = p_j\big(e^{t_j} - 1 - t_j\big) = p_j t_j\big(\rho(t_j) - 1\big). \tag{D.11}$$

Then $w_j \geq 0$, $R_s = \sum_j w_j(v_j - y)$, and

$$\sum_j w_j = \mathrm{KL}(p\|p') = \log D - \sum_j p_j \delta_j. \tag{D.12}$$

For a fixed downstream gradient g, put $a_j = \langle g, (v_j - y)W_o\rangle$ and $b_j = \langle g, \varepsilon_j W_o\rangle$. The error of the score/value linear response for the projected local proxy obeys

$$\begin{aligned}\big|\langle g, [(y' - y) - (L_s + V_0)]W_o\rangle\big| \leq\ & \mathrm{KL}(p\|p') \max_j |a_j| \\ & + \mathrm{TV}(p, p')\big(\max_j b_j - \min_j b_j\big),\end{aligned} \tag{D.13}$$

where $\mathrm{TV}(p, p') = \frac{1}{2}\sum_j |p'_j - p_j|$.

Proof. Convexity of the exponential gives $e^t - 1 - t \geq 0$. Since $\sum_j p_j e^{t_j} = 1$, the weights sum to $-\sum_j p_j t_j = \mathrm{KL}(p\|p')$. This bounds the contraction of R_s. The positive and negative parts of $p' - p$ each have mass $\mathrm{TV}(p, p')$; their contractions with b differ by at most that mass times the range of b. □

For multiple heads, apply the bound to each head with its own output projection and sum the bounds. The measured answer-margin prediction below retains a baseline downstream gradient; (D.13) certifies its local attention component.

The bound also certifies local score orderings. For candidate c, let $S_c = \langle g, \Delta y_c W_o\rangle$ be the exact local score, $\widehat{S}_c = \langle g, (L_{s,c} + V_{0,c})W_o\rangle$ its first-order (AtP) counterpart, and B_c a bound from (D.13). Then

$$\widehat{S}_a - \widehat{S}_b > B_a + B_b \quad \Longrightarrow \quad S_a > S_b. \tag{D.14}$$

Indeed, $S_a \geq \widehat{S}_a - B_a > \widehat{S}_b + B_b \geq S_b$. Likewise, $|\widehat{S}_c| > B_c$ certifies its sign. These statements concern the local scores under a fixed readout; they do not bound the remaining downstream nonlinearity. Evaluating the bound is not assumed cheaper than evaluating the exact response. Hard deletion uses its separate renormalisation formula below, rather than a finite-logit KL certificate.

D.3 SPARSE EVALUATION OF JOINT CACHE EDITS

Suppose scores change only on S and values only on T. With $r_j = p_j\delta_j\rho(\delta_j)$ for $j \in S$, the exact joint response is

$$y' - y = \frac{\sum_{j\in S} r_j(v_j - y) + \sum_{j\in T} p_j\varepsilon_j + \sum_{j\in S\cap T} r_j\varepsilon_j}{1 + \sum_{j\in S} r_j}. \tag{D.15}$$

Baseline probabilities, the readout, and candidate-group complement statistics can be shared across interventions. Each additional evaluation uses the changed entries. In particular, $\mathrm{KL}(p\|p') = \log D - \sum_{j\in S} p_j\delta_j$ and

$$2\,\mathrm{TV}(p,p') = \sum_{j\in S} p_j\left|e^{\delta_j}/D - 1\right| + \left(1 - \sum_{j\in S} p_j\right)\left|D^{-1} - 1\right|. \tag{D.16}$$

For evaluation, the denominator is the positive sum $\sum_{j\notin S} p_j + \sum_{j\in S} p_j e^{\delta_j}$, computed in log space with cached complement mass. Near complete attention concentration, the corresponding complement-weighted value sum also avoids subtracting two nearly equal readouts. These formulas support exact joint group evaluation and local error bounds from the same baseline capture.

D.4 FROM A POSITIONAL EDIT TO AN ANSWER-MARGIN PREDICTION

For a selected RoPE frequency band b, let A_b retain its blocks of A and set the other blocks to zero. Shifting a rotated key by δ gives

$$\widetilde{k}'_j = \widetilde{k}_j e^{\delta A_b^\top}, \qquad s'_j - s_j = \frac{\widetilde{q}\,\delta A_b\rho(\delta A_b)\widetilde{k}_j^\top}{\tau}. \tag{D.17}$$

Together, (D.17) and (D.2) retain both the finite rotary displacement and its softmax response. The positional-Jacobian comparator uses

$$\Delta y_{\mathrm{Jac}} = \sum_j p_j\delta_j^{\mathrm{lin}}(v_j - y), \qquad \delta_j^{\mathrm{lin}} = \frac{\widetilde{q}\,\delta A_b\widetilde{k}_j^\top}{\tau} \tag{D.18}$$

on the shifted keys, with $\delta_j^{\mathrm{lin}} = 0$ on the others.

Let $o_\ell = \sum_h y_{\ell h}W_{o,\ell h}$ be the combined attention write at the intervention site. For the two answer labels, fix the original model's preferred label r_* and define the signed logit margin $m = \mathrm{logit}(r_*) - \mathrm{logit}(1 - r_*)$. With $g_\ell = \nabla_{o_\ell} m$ at the original forward pass, compare

$$\widehat{\Delta m}_{\mathrm{fin}} = \langle g_\ell, \Delta o_{\ell,\mathrm{fin}}\rangle, \qquad \widehat{\Delta m}_{\mathrm{Jac}} = \langle g_\ell, \Delta o_{\ell,\mathrm{Jac}}\rangle. \tag{D.19}$$

The local increment in the first predictor is exact. Both predictors use the same baseline gradient to propagate that increment through the remaining network. The experimental target is the measured margin change $\Delta m = m' - m$ after executing the edit and continuing the model forward.

D.5 ADDITIVE ROTARY SCORES AND COUPLED SOFTMAX RESPONSES

Fix a query, its attended values, and a set of keys to be shifted. For a band b, let $d^b \in \mathbb{R}^N$ be the exact score displacement in (D.17), with zero displacement on the unshifted keys. Write $F(s) = \sum_j p_j(s)v_j$ for the head readout as a function of its scores.

Proposition D.2 (Softmax interaction between disjoint rotary bands)**.** *For disjoint rotary bands b, c, their joint score displacement satisfies $d^{b\cup c} = d^b + d^c$. The corresponding output interaction is*

$$\begin{aligned}\mathcal{H}_{bc} &= F(s + d^b + d^c) - F(s + d^b) - F(s + d^c) + F(s)\\ &= \int_0^1\int_0^1 D^2F(s + td^b + zd^c)[d^b, d^c]\,\mathrm{d}t\,\mathrm{d}z.\end{aligned} \tag{D.20}$$

For score directions $a, b \in \mathbb{R}^N$, the mixed derivative is

$$D^2F(s)[a,b] = \mathbb{E}_{p(s)}\left[(a - \mathbb{E}_{p(s)}a)(b - \mathbb{E}_{p(s)}b)(v - F(s))\right]. \tag{D.21}$$

Proof. Disjoint block supports give $e^{\delta(A_b+A_c)} - I = (e^{\delta A_b} - I) + (e^{\delta A_c} - I)$. Taking the query–key dot product proves score additivity. Applying the fundamental theorem of calculus in each score direction proves (D.20). Differentiating F twice and using $\partial p_j/\partial s_k = p_j(\mathbf{1}_{j=k} - p_k)$ yields (D.21). □

All four readouts retain the exact rotations. Thus (D.20) measures finite coupling through softmax between independent rotary score components. Output projection and summing heads preserve this decomposition by linearity.

D.6 Exact selection of cache entries for precision repair

For a single-token restoration, let δ_j be the score change from restoring its key and ε_j the change from restoring its value. With all other entries held fixed, (D.2) reduces to

$$\Delta_j y = \frac{p_j}{1 + p_j\delta_j\rho(\delta_j)} \left[\delta_j\rho(\delta_j)(v_j - y) + e^{\delta_j}\varepsilon_j\right]. \tag{D.22}$$

An equivalent implementation computes the new mass $p'_j = \mathrm{sigmoid}(\log p_j - \log(1-p_j) + \delta_j)$ and the complement average $y_{-j} = \sum_{k\neq j} p_k v_k/(1-p_j)$, giving $\Delta_j y = (p'_j - p_j)(v_j - y_{-j}) + p'_j\varepsilon_j$. For dominant tokens, the complement mass and average are recomputed with log-sum-exp over the other tokens.

Let $e = o_{\mathrm{corrupt}} - o_{\mathrm{original}}$ be the projected-write error. For candidate j, sum its head contributions before evaluating

$$B_j = \|e\|_2^2 - \|e + \Delta_j o\|_2^2, \qquad j_* \in \operatorname*{arg\,max}_{j \text{ not yet restored}} B_j. \tag{D.23}$$

The exact greedy procedure restores j_* and recomputes all candidate benefits at the new cache state. A restoration replaces the key and value for that physical token across all grouped-query attention KV groups. The original cache supplies the restoration values and the target write.

D.7 A symmetric secant for softmax

Equation (D.2) also has a matrix form. For the increments in (D.1), define

$$\lambda_j = p_j\rho(t_j) = \frac{p'_j - p_j}{\log p'_j - \log p_j}, \qquad \Lambda = \sum_j \lambda_j, \qquad \nu_j = \frac{\lambda_j}{\Lambda}. \tag{D.24}$$

Thus λ_j is the logarithmic mean of p_j and p'_j, equal to p_j when they coincide, and ν is a positive distribution on the attended set.

Proposition D.3 (Logarithmic-mean secant of softmax)**.** *For finite score increments δ on a fixed attended set,*

$$\log D = \nu^\top\delta, \qquad p' - p = \mathcal{S}(p,p')\,\delta, \qquad \mathcal{S}(p,p') = \mathrm{diag}(\lambda) - \frac{\lambda\lambda^\top}{\Lambda}. \tag{D.25}$$

The matrix $\mathcal{S}$ is symmetric and positive semidefinite, its kernel on the attended set is spanned by $\mathbf{1}$, it is unchanged when p and p' are exchanged, and it tends to the softmax Jacobian $\mathrm{diag}(p) - pp^\top$ as $\delta \to 0$. At fixed values,

$$K_f = \Lambda\,\mathrm{Cov}_\nu(\delta, v), \qquad L_s = \mathrm{Cov}_p(\delta, v). \tag{D.26}$$

Proof. By (D.2), $p'_j - p_j = \lambda_j t_j = \lambda_j(\delta_j - \log D)$. Summing over j gives $\log D = \nu^\top\delta$, and substituting back gives (D.25). For any vector a, $a^\top\mathcal{S}a = \Lambda\,\mathrm{Var}_\nu(a)$, which vanishes exactly when a is constant because every $\lambda_j > 0$. Exchanging p and p' leaves each logarithmic mean unchanged, and $\lambda \to p$ as $\delta \to 0$. Finally, $K_f = \sum_j (p'_j - p_j)v_j = \Lambda\sum_j \nu_j(\delta_j - \nu^\top\delta)v_j$, and (D.4) gives L_s in the same way. □

The identity $\log D = \nu^\top \delta$ is the Sato–Vartia index of index-number theory, which is exact for CES demand (Sato, 1976; Vartia, 1976); CES expenditure shares are a softmax of log prices. The componentwise relation $p'_j - p_j = \lambda_j(\log p'_j - \log p_j)$ is the logarithmic-mean chain rule of discrete optimal transport (Maas, 2011). Equation (D.26) puts the local responses on one scale. Attribution patching evaluates the covariance of score increments with values under the baseline distribution; the exact finite key response K_f, which the AtP* QK fix contracts, evaluates it under the logarithmic-mean distribution ν and scales it by Λ. Their difference is the remainder R_s of (D.10), whose weights sum to $\mathrm{KL}(p\|p')$; for positional edits it is the softmax remainder of (7), which changes the measure from p to ν and the scale from 1 to Λ. With simultaneous value edits, the midpoint rule of Appendix D.8 writes the exact response as $\Lambda \operatorname{Cov}_\nu(\delta, \bar{v}) + \sum_j \bar{p}_j \varepsilon_j$, with $\bar{v}_j = v_j + \varepsilon_j/2$ and $\bar{p}_j = (p_j + p'_j)/2$. This symmetric split assigns $K_f + \frac{1}{2}C_{KV}$ to the keys and $V_0 + \frac{1}{2}C_{KV}$ to the values; (D.7) keeps the interaction as its own term.

Taking uniform attention as the baseline, with $p_j = 1/N$ and $\delta = s$, gives the readout itself: $y - \mu = \Lambda \operatorname{Cov}_\nu(s, v)$, where μ is the uniform value mean and λ_j is the logarithmic mean of $1/N$ and the attention weight e^{s_j}/Z. These λ_j are the effective coefficients c_j^g of (C.2) in the gauge $\sigma = \log(Z/N)$, which is $\nu^\top s$ by (D.25). The gradient-step representation (11) and the finite response are therefore one identity taken from two baselines: uniform attention for the effective matrix, and the unedited head for an intervention.

D.8 Exact propagation through the remaining network

The margin predictions (D.19) and (F.1) contract an exact local response with the baseline gradient. The divided-difference calculus of Theorem 1 also propagates that response exactly. For a map F and two states z, z', a linear operator $\mathcal{F}(z, z')$ is an *endpoint secant* of F when $F(z') - F(z) = \mathcal{F}(z, z')(z' - z)$. Every secant below is symmetric in (z, z') and equals the Jacobian of F at z when $z' = z$. Token states are column vectors, and bars denote endpoint means.

Elementary secants. For any bilinear map B,

$$B(a', b') - B(a, b) = B(a' - a, \bar{b}) + B(\bar{a}, b' - b), \tag{D.27}$$

which covers attention scores, attention readouts, and gating products. For RMS normalisation $N(x) = \operatorname{diag}(\gamma)x/r + \beta$ with $r = (\|x\|_2^2/d + \epsilon)^{1/2}$, applying (D.27) to $x \cdot r^{-1}$ and using $r' - r = (x + x')^\top(x' - x)/[d(r + r')]$ gives

$$N(x') - N(x) = \operatorname{diag}(\gamma)\left[\frac{r + r'}{2rr'}I - \frac{2\bar{x}\bar{x}^\top}{d\, rr'(r + r')}\right](x' - x). \tag{D.28}$$

LayerNorm applies (D.28) to Cx and Cx', with $C = I - \mathbf{1}\mathbf{1}^\top/d$, and right-multiplies by C. An elementwise activation ϕ has the diagonal secant $\mathcal{D}_\phi$ with entries $[\phi(a'_k) - \phi(a_k)]/(a'_k - a_k)$, or $\phi'(a_k)$ when $a'_k = a_k$. For a gated MLP $M(x) = W_d[\phi(a) \odot b] + b_d$ with $a = W_a x + b_a$ and $b = W_b x + b_b$, (D.27) gives the secant

$$\mathcal{M} = W_d\big[\operatorname{diag}(\bar{b})\, \mathcal{D}_\phi W_a + \operatorname{diag}\big(\overline{\phi(a)}\big) W_b\big]. \tag{D.29}$$

Attention. Fix positions, masks, and each row's attended set; cached entries enter as constants. For normalised inputs X, X' of an attention sublayer, the rotated queries and keys and the values change linearly with $X' - X$. By (D.27), the score increments are $\Delta S = (\Delta\widetilde{Q}\,\widetilde{\bar{K}}^\top + \widetilde{\bar{Q}}\,\Delta\widetilde{K}^\top)/\tau$; Proposition D.3 gives $\Delta p_i = \mathcal{S}(p_i, p'_i)\,\Delta s_i$ on each row; and (D.27) gives $\Delta Y = \Delta P\,\bar{V} + \bar{P}\,\Delta V$. With these endpoint coefficients fixed, the composition with output projection and head summation is a linear operator $\mathcal{A}$ satisfying $\mathrm{Attn}(X') - \mathrm{Attn}(X) = \mathcal{A}(X' - X)$.

Theorem D.4 (Exact secant factorisation of a transformer)**.** *Consider pre-norm blocks $u_\ell = h_\ell + A_\ell(N_{\ell,1}(h_\ell))$ and $h_{\ell+1} = u_\ell + M_\ell(N_{\ell,2}(u_\ell))$ for $\ell < L$, with readout $z = W_U N_f(h_L) + b_U$ and fixed attended sets. Let $\mathcal{N}_{\ell,1}$ and $\mathcal{N}_{\ell,2}$ be the normalisation secants at (h_ℓ, h'_ℓ) and (u_ℓ, u'_ℓ), let $\mathcal{A}_\ell$ and $\mathcal{M}_\ell$ be the sublayer secants at the corresponding normalised states, let $\mathcal{N}_f$ be the final normalisation secant, and put $\mathcal{T}_\ell = (I + \mathcal{M}_\ell\mathcal{N}_{\ell,2})(I + \mathcal{A}_\ell\mathcal{N}_{\ell,1})$. Then $\Delta h_{\ell+1} = \mathcal{T}_\ell \Delta h_\ell$. If an intervention first changes the residual stream through the attention write of layer ℓ_0, then*

$$\Delta z = W_U \mathcal{N}_f\, \mathcal{T}_{L-1} \cdots \mathcal{T}_{\ell_0+1}\, (I + \mathcal{M}_{\ell_0}\mathcal{N}_{\ell_0,2})\, \Delta u_{\ell_0}. \tag{D.30}$$

Every factor is unchanged when the baseline and edited trajectories are exchanged, and reduces to the corresponding Jacobian when they coincide.

Proof. The secant identities give $A_\ell(N_{\ell,1}(h'_\ell)) - A_\ell(N_{\ell,1}(h_\ell)) = \mathcal{A}_\ell \mathcal{N}_{\ell,1} \Delta h_\ell$ and the corresponding MLP equality at (u_ℓ, u'_ℓ); the residual connections add the identity. Induction over blocks and the affine readout give (D.30). Symmetry and the Jacobian limit hold factor by factor. □

The finite gradient. For a scalar margin $m = w^\top z$, transposing (D.30) gives

$$g_{\mathrm{fin}}^\top = w^\top W_U \mathcal{N}_f \, \mathcal{T}_{L-1} \cdots \mathcal{T}_{\ell_0+1} \, (I + \mathcal{M}_{\ell_0} \mathcal{N}_{\ell_0,2}), \qquad \Delta m = \langle g_{\mathrm{fin}}, \Delta o_{\ell_0} \rangle, \tag{D.31}$$

where $\Delta o_{\ell_0} = \Delta u_{\ell_0}$ is the projected local response of Theorem 1, whose key term is given by (D.26). As the edit shrinks, every factor tends to its Jacobian and $g_{\mathrm{fin}} \to g_{\ell_0}$. The frozen-gradient prediction therefore differs from the executed change by exactly

$$\Delta m - \widehat{\Delta m} = \langle g_{\mathrm{fin}} - g_{\ell_0}, \Delta o_{\ell_0} \rangle. \tag{D.32}$$

The local factor is exact, so the executed error of every finite predictor in Section 5 is propagation error. The comparators add a local term to the same propagation term. With projected responses Δo^{AtP} and $\Delta o^{\mathrm{AtP}^*}$ from $L_s + V_0$ and $K_f + V_0$,

$$\begin{aligned} \Delta m - \langle g_{\ell_0}, \Delta o^{\mathrm{AtP}^*} \rangle &= \langle g_{\ell_0}, C_{KV} W_o \rangle + \langle g_{\mathrm{fin}} - g_{\ell_0}, \Delta o_{\ell_0} \rangle, \\ \Delta m - \langle g_{\ell_0}, \Delta o^{\mathrm{AtP}} \rangle &= \langle g_{\ell_0}, (R_s + C_{KV}) W_o \rangle + \langle g_{\mathrm{fin}} - g_{\ell_0}, \Delta o_{\ell_0} \rangle, \end{aligned} \tag{D.33}$$

with heads summed after projection. The local terms are bounded by (D.13), and the interaction term is the quantity whose contraction tracks the executed four-cell interaction in Section 5. Evaluating the transposed chain at an intermediate layer gives $\Delta m = \langle g_{\mathrm{fin},\ell}, \Delta h_\ell \rangle$ for every $\ell > \ell_0$, which attributes an executed edit exactly to each downstream residual coordinate and, through the sublayer factors, to each head and MLP write. The factors use the edited trajectory, which execution supplies for every selected candidate. An edit at several readouts adds a source at each edited node, as in $\Delta(Rx) = R' \Delta x + (R' - R)x$, and the same factors propagate every source.

Numerical check. A float64 implementation on a three-block model with causal RoPE attention, RMSNorm, SwiGLU MLPs, and final normalisation reproduces executed logit changes to within 2.4×10^{-15} over 25 residual-stream interventions whose scales span more than three orders of magnitude. For finite key rotations at an intermediate readout, the covariance form (D.26) reproduces the executed attention write to 7.2×10^{-16}, and propagating it through (D.30) reproduces the logits to 1.3×10^{-15}. Exchanging the two trajectories changes the materialised operator by at most 1.1×10^{-15}.

Relation to prior constructions. Finite-difference multipliers composed by a chain rule define DeepLIFT (Shrikumar et al., 2017), whose Rescale rule is the activation secant above and which acts as a modified gradient (Ancona et al., 2018); slope arithmetic gives the general composition rule (Schichl & Neumaier, 2005), and (D.27) is the classical two-factor decomposition of Bennet (1920). Integrated gradients reach an exact finite gradient by path integration (Sundararajan et al., 2017). Relevance-propagation attributions for transformers linearise softmax or hold attention and normalisation fixed (Ali et al., 2022; Achtibat et al., 2024), as do exact reconstructions at a single input (Golden, 2025; Ameisen et al., 2025). Concurrently, the tangermeme library released closed-form DeepLIFT rules for LayerNorm, RMSNorm, softmax, and bilinear contractions (Schreiber, 2026); its softmax multiplier is exact on the realised increment but asymmetric, and it does not annihilate the common-logit direction. Here each factor follows from the divided-difference identity of Theorem 1, softmax uses the symmetric secant of Proposition D.3 with its covariance form, the exact RoPE response enters as the source, and (D.32) gives the downstream error of attribution patching in closed form. Zhang & Wang (2026) attribute that error mainly to the network's response to the intervention and correct its leading order with a Hessian–vector product.

E PRETRAINED EFFECTIVE-WEIGHT RECONSTRUCTION

We evaluate Qwen2.5-0.5B (Qwen Team, 2024b), layer 23 (zero-indexed), on $C = 16$ article prefixes of 511 tokens, all 14 query heads, and $Q = 32$ alternative one-token continuations at position 511.

Table E.1: Algebraic reconstruction on SmolLM2-1.7B Base. Both the effective-weight and standard-attention paths use binary128. Errors are measured between their combined projected outputs over all evaluated contexts and continuations.

Readout	MSE	Maximum absolute error
Prefix-only	3.37×10^{-49}	1.84×10^{-22}
Full causal	1.92×10^{-49}	1.81×10^{-22}

Continuation 0 is the observed next token; the other 31 are the model's highest-ranked distinct nonspecial alternatives. Each continuation reads the same prefix bank. Projection biases and the model's pre-normalised head inputs are included. Reconstruction is checked both for the prefix-only readout and for the full causal readout including the continuation's own key/value.

For the latter, if the self token receives mass π_i, the exact relation is $y_i^{\text{full}} = (1 - \pi_i) y_i^{\text{prefix}} + \pi_i v_i^{\text{self}}$. The full-bank version of (B.6) uses its own scores and uniform mean. The 240 reconstruction rows comprise 224 individual head/context pairs and 16 combined projected writes. All pass the reported checks. Combined prefix and full reconstruction MSEs in float64 are approximately 9.8×10^{-30} and 9.6×10^{-30}. Against the native FP32 combined write, the MSE is 7.01×10^{-14} and the maximum absolute difference is 1.4×10^{-5}. The float64 figures measure agreement between algebraically equivalent evaluations; the native comparison measures agreement at the model's execution precision.

The exact tokenwise correction in (C.3) also expresses the change between a reference effective matrix and the matrix for a new query. Adding that correction to the reference readout recovers the exact attention output. Together, the projection-coordinate and cache-coordinate evaluations verify the construction on the pretrained architecture.

Cross-model algebraic reconstruction. We extend the algebraic reconstruction check to SmolLM2-1.7B Base at layer 23 (zero-indexed), using 16 article prefixes of 511 tokens, all 32 query heads, and 32 alternative one-token continuations per prefix. The effective-weight construction and standard-attention reference use identical captured inputs and model weights. Some samples exhibit numerical cancellation amplified by a small original-score normalization constant; we therefore evaluate both paths in IEEE binary128 arithmetic. This precision applies to the reconstruction calculations, not the model forward pass.

All 512 head/context algebraic readout checks and 16 combined projected-write checks satisfy the original tolerances. Table E.1 reports agreement for both the prefix-only readout and the full causal readout, including the continuation's own key/value. The anchor-plus-tokenwise-correction readouts also satisfy the prescribed tolerances.

F JOINT KEY–VALUE ATTRIBUTION IN PRETRAINED ATTENTION

We test whether the exact local response predicts executed downstream changes when contracted with one baseline gradient. The fixed supplemental sweep covers six model/task/layer settings and 5,120 executed cache interventions. It also measures the interaction in (D.9), sensitivity to edit strength, and the cost of sparse evaluation. Appendix J retains the initial single-layer study.

F.1 PAIRED PROMPTS AND EXECUTED CACHE INTERVENTIONS

We use frozen Qwen2.5-0.5B-Instruct at zero-indexed layer 18 and Qwen2.5-1.5B-Instruct (Qwen Team, 2024a) at layers 14 and 21. Each model/layer combination is evaluated on entity–label retrieval and SST-2 sentiment prompts (Socher et al., 2013). All six settings, 32 prompt pairs per task, and eight candidate spans per prompt are fixed before model evaluation. The same task prompts are used across the selected model/layer combinations, giving 64 distinct prompt pairs and 192 setting–pair evaluations. No prompt is filtered by model accuracy or intervention outcome.

A retrieval donor cyclically permutes entity identifiers and flips their A/B labels. An SST-2 prompt contains eight balanced labelled training examples and a held-out validation query; its donor flips the demonstration labels while preserving the review text and query. Native tokenisation checks equal

endpoint lengths, aligned intervention spans, and unchanged token IDs outside those spans. Retrieval prompts have 112 tokens; SST-2 prompts have 204–300 tokens, including the native chat template. Appendix K records the pinned checkpoints, prompt construction, and execution protocol.

For every span, we execute donor key-only, value-only, and joint replacements across all physical KV groups at the selected layer. Each continuation receives an independent copy of the baseline prefix cache. The baseline query, its self entry, and all unpatched entries are retained. Four patched continuations share a batch with one unpatched control. The target is the fixed answer margin $m = \mathrm{logit}(A) - \mathrm{logit}(B)$; the measured effect subtracts the control margin from the patched margin in the same batch. Full replacement gives $6 \times 32 \times 8 \times 3 = 4{,}608$ patched continuations. At 1.5B retrieval layer 21, joint edits of strengths $\alpha = 0.1$ and 0.5 add 512 continuations, for 5,120 in total.

F.2 Matched predictors from the finite-change decomposition

At the selected layer ℓ, let $\mathcal{O}$ project and sum the head outputs. Define the common linear readout $\mathcal{G}(a) = \langle g_\ell, \mathcal{O}(a) \rangle$, using one baseline gradient for all candidates and predictors. Equation (D.9) then supplies the proposed prediction

$$\widehat{\Delta m}_{\mathrm{joint}} = \mathcal{G}\big((\mu' - \mu) + u^\top(\Delta M' - \Delta M)\big) = \mathcal{G}(K_f + V_0 + C_{KV}). \tag{F.1}$$

The change in the uniform mean is included. Computation uses the factored cache-coordinate response in (D.15), with positive group and complement masses evaluated in log space.

The score/value first-order predictor, attribution patching (AtP) on the edited keys and values (Syed et al., 2024), uses $\mathcal{G}(L_s + V_0)$. The AtP* predictor uses $\mathcal{G}(K_f + V_0)$, combining the exact grouped fixed-value key response with the value response at baseline probabilities. This is the QK fix of AtP* (Kramár et al., 2024) applied to the whole span as one node, with values patched linearly. A quadratic interaction comparison retains the same exact key response and adds

$$C_2 = \sum_j p_j \left(\delta_j - \sum_k p_k \delta_k \right) \varepsilon_j, \qquad \widehat{\Delta m}_{\mathrm{quad}} = \mathcal{G}(K_f + V_0 + C_2). \tag{F.2}$$

Along the joint path $(\alpha\delta, \alpha\varepsilon)$, $C_{KV}(\alpha) = \alpha^2 C_2 + O(\alpha^3)$. Thus this comparison retains the leading mixed term. All predictors evaluate the same complete span and share the same gradient; none fits a parameter. Direct dense attention recomputation, followed by $\mathcal{G}$, supplies an independent equality control for the factored joint response.

F.3 Prediction accuracy and influential-span recovery

For prompt pair u and candidate span j, write $e^a_{uj} = |\widehat{\Delta m}^a_{uj} - \Delta m_{uj}|$. For each setting, the AtP* contrast is

$$\Delta_{\mathrm{MAE}} = \frac{1}{32} \sum_{u=1}^{32} \left[\frac{1}{8} \sum_{j=1}^{8} \left(e^{\mathrm{joint}}_{uj} - e^{\mathrm{AtP}^*}_{uj} \right) \right]. \tag{F.3}$$

We form its percentile 95% interval using 5,000 paired bootstrap resamples of the 32 prompt pairs, keeping all eight joint patches together. Top-two recall is the overlap between the two spans ranked highest by absolute predicted effect and the two ranked highest by absolute executed effect, divided by two and averaged over prompts. Ties use span order. Sign accuracy compares signed predictions with nonzero executed margin changes. Sign accuracy and top-two recall are undefined for the zero predictor. We compute the same paired MAE contrast against the zero, AtP, and quadratic predictors. All intervals are individual 95% intervals, without multiplicity adjustment.

Figure F.1 shows full-replacement MAE for every setting. The exact joint predictor has the lowest MAE in all six, reducing error by 75.9–98.5% against the quadratic interaction correction, by 89.3–99.6% against AtP*, and by 92.7–99.6% against AtP. All 24 setting/comparator paired intervals favor exact joint. The complete means and the AtP* and quadratic paired contrasts appear in Tables K.1 and K.2; the report archive retains all four contrasts.

At 1.5B retrieval layer 21, the exact, AtP*, and quadratic MAEs are 0.007844, 0.107864, and 0.087683, respectively. At 1.5B SST-2 layer 14, where the relative gain against the quadratic correction is smallest, they are 0.000342, 0.003194, and 0.001418. Full-replacement top-two recall is 98.44–100%

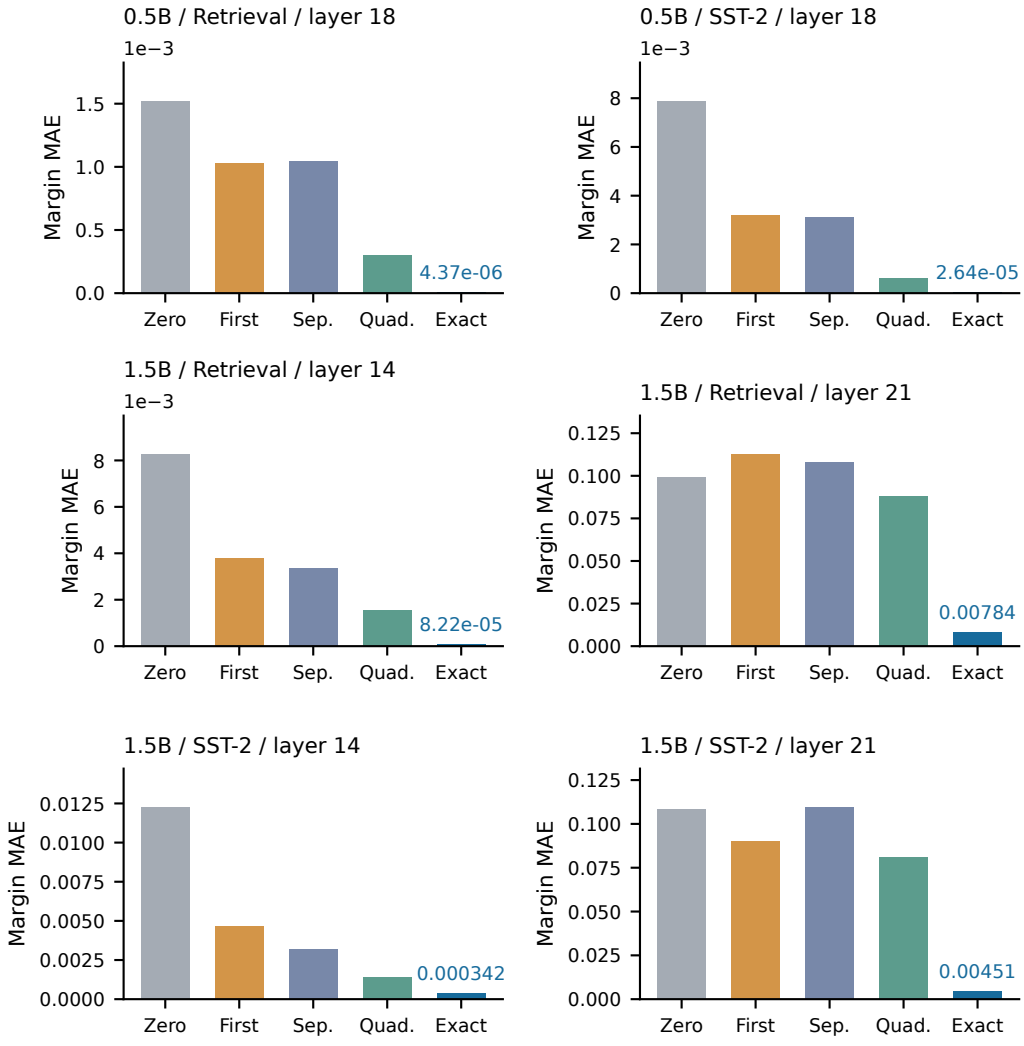

Figure F.1: Downstream prediction across six fixed settings at full replacement. Bars plot the reported margin MAEs over 32 prompt pairs with eight joint span edits each (Table K.1); First and Sep. denote AtP and AtP*. Axes are scaled separately, and exact-joint MAEs are labelled. All predictors share the same baseline downstream gradient; targets are independently executed margin changes. The quadratic comparator retains the exact key response and leading mixed term. Paired 95% intervals are reported in Table K.2.

and sign accuracy is 99.22–100% for exact joint. The zero predictor is included throughout: AtP* exceeds its MAE in both 1.5B layer-21 settings, whereas exact joint improves on zero in every setting.

F.4 Executed interaction and intervention strength

Key-only and value-only executions define an additional target:

$$I_m = (m_{KV} - m_0) - (m_K - m_0) - (m_V - m_0), \qquad \widehat{I}_m = \mathcal{G}(C_{KV}). \tag{F.4}$$

Across the six settings, the signed predicted interaction has Pearson correlation 0.9970–0.99999 with the executed four-cell effect. Interaction MAEs range from 4.52×10^{-6} to 0.008826, versus 2.96×10^{-4} to 0.092810 for the quadratic mixed term (Table K.3). The common baseline gradient propagates each local interaction to a prediction of the independently executed four-cell effect.

For the strength sweep, rotated keys and values follow $k'_j = k_j + \alpha(k_j^{\text{donor}} - k_j)$ and $v'_j = v_j + \alpha(v_j^{\text{donor}} - v_j)$ on the selected span. Prediction and execution use the same rounded FP32 endpoint tensors. At $\alpha = 0.1, 0.5, 1.0$, exact-joint MAE is respectively 0.000173, 0.002330, and 0.007844; quadratic MAE is 0.000237, 0.013071, and 0.087683 (Figure F.2). At $\alpha = 0.1$, the quadratic and exact predictors both attain 100% top-two recall. The small absolute MAE difference, approximately -6.34×10^{-5} for exact minus quadratic, has paired 95% interval $[-1.09 \times 10^{-4}, -2.88 \times 10^{-5}]$. The gap grows with intervention strength, while exact-joint MAE also grows as the downstream approximation becomes less accurate.

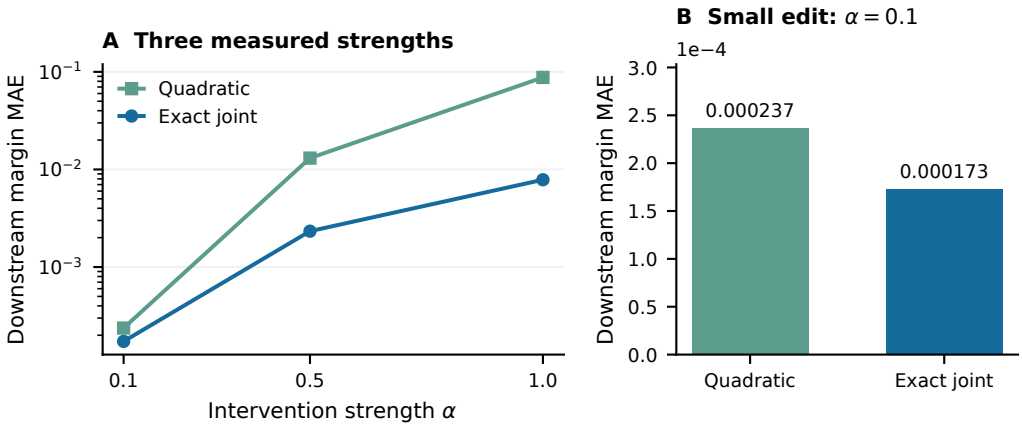

Figure F.2: Intervention-strength diagnostic on 1.5B retrieval layer 21, using the same 32 prompt pairs and eight spans at each strength. **A:** Exact-joint and quadratic mean downstream errors at the three evaluated strengths; lines connect measured points on a log scale. **B:** The same predictors at $\alpha = 0.1$ on a linear scale. The plot uses reported summary means. The paired 95% interval for their small-strength difference is $[-1.09 \times 10^{-4}, -2.88 \times 10^{-5}]$.

F.5 Numerical audit and scope

The supplemental run passes all 14 CPU and 14 CUDA preflight test methods, including the exact affine-input gradient-step identity, native grouped-query replay, cache isolation, and finite-difference checks of the projected-write gradient. Across all 192 setting–pair evaluations, the largest recorded FP64 sparse–dense local discrepancy is 1.31×10^{-14}. The maximum difference between the predicted local contraction and its executed FP32 counterpart is 2.60×10^{-6}. The maximum unpatched control-margin drift is 1.14×10^{-5}; each intervention uses its own batch control. All 5,120 intervention records are present, with no duplicate setting/pair/span/kind/strength combinations. The smallest downstream MAEs approach the FP32 numerical scale; they should not be interpreted as exact downstream predictions.

The coverage spans two model sizes within the Qwen family and two prompt tasks. Baseline A/B accuracy is 46.88% on retrieval and 59.38% on SST-2 for 0.5B, and 96.88% and 87.50%, respectively,

for 1.5B. These accuracies are reported as task context; the evaluated quantity is prediction of intervention effects. The finite-response identity is exact for every permitted head and edit, while the measured downstream accuracy pertains to these checkpoints, layers, and prompts.

F.6 Runtime with preparation and reuse

We compare sparse evaluation of (D.15) with batched dense attention recomputation for the same requested output: one scalar response per candidate after contraction with the fixed baseline gradient. Both methods receive identical FP32 inputs and may precontract values with that gradient. Dense evaluation recomputes the full softmax for every candidate; it batches all candidates and shares the unchanged value bank. Prefix/model execution and baseline-gradient acquisition are common inputs and are excluded from both timings. The measured cost is local response evaluation, rather than full-model inference latency.

The native timing case uses the first saved 1.5B retrieval layer-21 capture with 112 tokens and eight seven-token candidate spans. Six synthetic cases cross $N \in \{128, 1024, 8192\}$ context tokens with $C \in \{8, 256\}$ candidates and eight-token spans, using 12 query heads, two KV heads, and head width 128. All cases use an NVIDIA A100-SXM4-40GB, warmed and synchronised timing, and five rounds with randomised method order. We report medians and interquartile ranges. The setup-inclusive measurements charge common preparation and, for sparse evaluation, candidate-group complement preparation. The latter conservatively scans the bank for each candidate; it is not treated as free preprocessing.

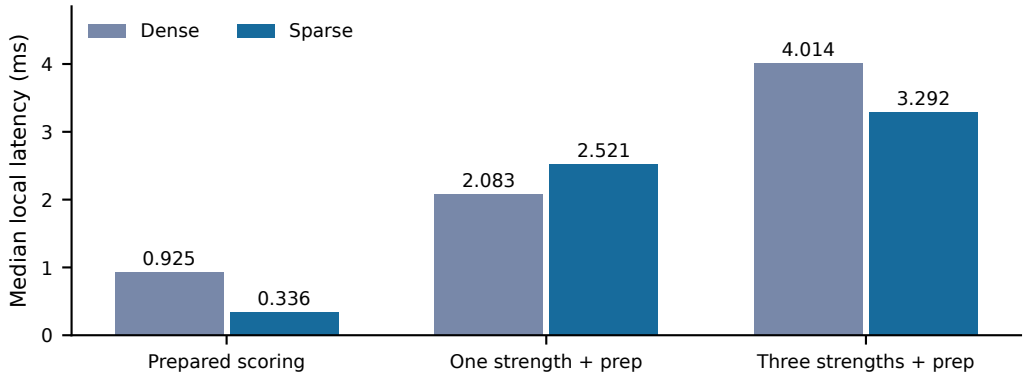

Figure F.3: Native local-response latency for the same eight candidates on an A100. Bars show the reported medians over five timing rounds. Scoring-only latency assumes prepared statistics. The other groups directly time preparation plus evaluation at one or three strengths; three strengths share one preparation in both methods. Model execution and gradient acquisition are excluded throughout. Both dense and sparse methods evaluate the exact local response.

On the native case, scoring with prepared statistics takes 0.336 ms for sparse evaluation and 0.925 ms for dense recomputation, a 2.75× scoring speedup. Including preparation, one strength takes 2.521 ms versus 2.083 ms, making sparse evaluation 21.0% slower. Three strengths sharing preparation take 3.292 ms versus 4.014 ms, an 18.0% latency reduction (1.22× speedup). Across all seven cases, one-strength sparse evaluation is 18.2–21.0% slower, while three-strength reuse reduces latency by 18.0–21.8% (1.22–1.28×; Table K.6). Thus the implementation benefits from reusing group statistics, with a measured preparation cost. These results describe this eager implementation and hardware; they do not establish a speed advantage over fused attention kernels.

G Positional prediction, decomposition, and precision repair

G.1 Protocol, tasks, and statistical units

We evaluate the same frozen Qwen2.5-1.5B-Instruct checkpoint as Section F, with its chat template, FP32 eager attention, and grouped-query attention (12 query heads, 2 KV groups). Calibration

examines layers $\{0, 7, 14, 21, 27\}$, using 32 disjoint prompt sets from each of four families: decisive-attribute inference (calibration only), corrupted category mapping, ordered register assignments, and SST-2 sentiment. The shared calibration policy is fixed before the test evaluations reported here.

The Qwen held-out evaluation comprises 256 prompt sets per task, 768 in total: *Corrupt* asks for a two-category mapping from eight demonstrations with one flipped label; *Ordered* asks for the last assignment to a target register; *SST-2* uses eight training-set demonstrations, one label flipped, and a held-out sentiment query from the validation split (Socher et al., 2013). SST-2 query examples and demonstration supports are sampled without reuse across prompt sets, including calibration versus test. Each prompt set has four demonstration orders and a fifth variant that appends 16 repetitions of an irrelevant note before the query. Ordered-task labels follow the actual assignment order. Calibration and test use separate deterministic streams with seed 24117.

Prediction errors and sign agreements are averaged within each prompt set and then across the 256 sets for each task. Paired comparisons use the same prompt sets for both methods. Reported 95% intervals are pointwise percentile intervals. Positional prediction summaries use 5,000 paired prompt-level bootstrap resamples; frequency-interaction and repair summaries use 2,000. Multiple interventions and prompt variants remain within their original prompt cluster. Answer NLL and KL are computed on the model distribution conditional on the two labels A/B.

G.2 Predicting the effects of finite positional interventions

At layers $\{0, 14, 21\}$ retained by the shared calibration policy, we apply shifts of 16 and 128 positional units to the cached keys of the demonstration tokens. The 64 rotary planes are sorted by frequency and partitioned into bands of 22 fast, 21 middle, and 21 slow planes; an additional intervention shifts all planes. Values, causal masks, and the current query are fixed at the edited readout. Each edit is executed through the remaining model to measure its downstream effect. The design supplies $5 \times 3 \times 2 \times 4 = 120$ interventions per prompt set and 30,720 per task. All predictors use the same interventions and baseline downstream gradient.

Table G.1 compares (D.19) against the measured answer-margin changes. The zero predictor has $\widehat{\Delta m} = 0$, so its MAE is the mean magnitude of the observed response.

Table G.1: Prediction of executed positional edits on 256 held-out prompt sets per task. MAE is in downstream logit-margin units. Reductions compare finite and zero MAE; brackets give paired 95% prompt-bootstrap intervals for the percentage reduction.

Task	Finite MAE	Jacobian MAE	Zero MAE	Reduction vs. zero
Corrupt	0.02808	0.16028	0.05724	50.9% [48.2, 53.7]
Ordered	0.05786	0.21897	0.09073	36.2% [35.0, 37.5]
SST-2	0.03585	0.15426	0.06681	46.3% [43.1, 49.9]

Finite prediction reduces MAE by 36.2–50.9% relative to zero and 73.6–82.5% relative to the positional Jacobian. Finite-predictor sign accuracy is 96.52% on Corrupt, 95.36% on Ordered, and 96.40% on SST-2; corresponding Jacobian accuracies are 75.01%, 74.32%, and 73.12%. Paired finite-minus-zero MAE intervals are $[-0.03040, -0.02789]$, $[-0.03401, -0.03174]$, and $[-0.03291, -0.02906]$, respectively.

G.3 Positional validation on SmolLM2

We additionally evaluate SmolLM2-1.7B-Instruct at revision `31b70e2e869a7173562077fd711b654946d38674`. The experiment retains the task construction, intervention definitions, and prompt-level statistical units described above, using the checkpoint's native tokenizer and chat template. SST-2 is pinned to revision `8d51e7e4887a4caaa95b3fbebbf53c0490b58bbb`.

Calibration uses 32 prompt sets from each of the four task families and examines layers $\{0, 6, 12, 18, 23\}$. The jointly selected positional policies retain layers $\{12, 18\}$ for held-out evaluation. Testing uses 256 prompt sets each for Corrupt, Ordered, and SST-2. Five variants, two retained layers, two shifts, and four band choices give 80 positional interventions per prompt set and 61,440 in total. The 32 rotary planes are partitioned into 11 fast, 11 middle, and 10 slow planes.

Finite prediction reduces Jacobian MAE by 98.27%, 97.22%, and 98.24% on Corrupt, Ordered, and SST-2, respectively. Corresponding reductions against zero are 88.53%, 86.49%, and 91.99%, and sign accuracies are 98.11%, 98.60%, and 98.51%. All task-wise paired 95% intervals for finite-minus-comparator MAE lie below zero for both comparators. Intervals use 5,000 resamples of complete prompt sets (Table G.2). Model-specific tokenization, rotary frequencies, and calibration-selected layers differ between the two evaluations. We therefore interpret the results as within-model comparisons of predictors, rather than as a ranking of the models by raw MAE.

Table G.2: SmolLM2 positional prediction on 256 held-out prompt sets per task. MAE differences are finite minus comparator, with paired 95% prompt-bootstrap intervals; negative values favour finite prediction. Sign accuracy is for the finite predictor. Absolute MAEs appear in Table 1.

Task	Δ_{Jacobian}	95% interval	Δ_{zero}	95% interval	Sign accuracy
Corrupt	-0.183064	[-0.191090, -0.175368]	-0.024922	[-0.025644, -0.024210]	98.11%
Ordered	-0.257617	[-0.261724, -0.253390]	-0.047099	[-0.048125, -0.046013]	98.60%
SST-2	-0.132747	[-0.138094, -0.127684]	-0.027278	[-0.028269, -0.026294]	98.51%

G.4 Frequency, displacement, and layer resolve the response

The angular displacement of plane r is $\theta_r = \delta\omega_r$. The saved Qwen frequencies give the following scales:

Table G.3: Maximum absolute rotary angle in each frequency band, in radians.

Band	Shift 16	Shift 128
Fast	16.000	128.000
Middle	0.1386	1.1084
Slow	0.001489	0.011911

At middle-band shift 16, the Jacobian reduces zero-predictor MAE by 94.0–94.8%. The finite response further reduces its remaining error by 72.9–85.9% (Table G.4). At shift 128 the middle-band response has mean magnitude 0.028–0.039; the finite predictor reduces Jacobian error by 80.2–92.0%. Each row averages 3,840 interventions from 256 prompts.

Table G.4: Middle-band prediction across two finite displacements. Absolute errors and the zero baseline specify the response scale; the final column compares finite and Jacobian MAE.

Task	Shift	Finite MAE	Jacobian MAE	Zero MAE	Reduction
Corrupt	16	0.000064	0.000236	0.004576	72.9%
Ordered	16	0.000057	0.000402	0.007160	85.9%
SST-2	16	0.000040	0.000241	0.004042	83.4%
Corrupt	128	0.003543	0.017862	0.031212	80.2%
Ordered	128	0.002239	0.027925	0.038921	92.0%
SST-2	128	0.002112	0.016428	0.027545	87.1%

The paired 95% intervals for the shift-16 reductions over the Jacobian are $[71.1, 74.6]\%$, $[85.4, 86.3]\%$, and $[82.0, 84.7]\%$ on Corrupt, Ordered, and SST-2. In the slow band at shift 16, both predictors have MAE between 4×10^{-6} and 6×10^{-6} at native FP32 execution precision, exhibiting their agreement in the small-angle regime. The full band-by-shift means appear in Appendix L.

Layer stratification further locates accurate downstream propagation. At layer 14, finite prediction reduces zero-predictor error by 80.1%, 90.9%, and 86.0% on Corrupt, Ordered, and SST-2; at layer 21 the reductions are 94.7%, 93.0%, and 92.0%. These summaries use every recorded band and shift at the stated layer.

To describe responses with larger measured magnitude, we also stratify the recorded interventions by $|\Delta m| \geq 0.1$. This yields 4,285 Corrupt, 6,201 Ordered, and 4,195 SST-2 interventions, with all 256

prompts represented in each task. Finite-predictor MAE is 42.2%, 26.2%, and 30.2% below zero, and its sign accuracy is 90.9%, 85.6%, and 89.0%, respectively. Each prompt is weighted equally after averaging its interventions in this response stratum.

G.5 Measuring softmax coupling between rotary bands

We evaluate the four local readouts in (D.20) on the same pretrained query, cache, demonstration-token mask, and output projection. The two bands and their union use exact rotations of the original keys. For prompt u and variant v, define the projected interaction and joint effect

$$H_{uv}^{bc} = \sum_h \mathcal{H}_{bc,h} W_{o,h}, \qquad T_{uv}^{bc} = \sum_h \big[F_h(s + d^b + d^c) - F_h(s)\big] W_{o,h}, \tag{G.1}$$

and report the mean relative interaction norm

$$\overline{\kappa}_{bc} = \frac{1}{256} \sum_u \frac{1}{5} \sum_{v=1}^{5} \frac{\big\|H_{uv}^{bc}\big\|_2}{\|T_{uv}^{bc}\|_2}. \tag{G.2}$$

Heads are summed after projection and before either norm is taken.

Table G.5: Fast–middle softmax interaction as a percentage of the combined projected local response, using (G.2). Each cell contains 256 prompt sets and five variants. Brackets give 95% prompt-bootstrap intervals.

Task	Layer	Shift 16	Shift 128
Corrupt	0	9.20 [9.11, 9.28]	23.92 [23.21, 24.58]
Corrupt	14	3.33 [3.30, 3.36]	18.96 [18.91, 19.01]
Corrupt	21	4.78 [4.73, 4.82]	17.22 [17.06, 17.41]
Ordered	0	5.31 [5.31, 5.32]	26.94 [26.86, 27.02]
Ordered	14	5.61 [5.59, 5.63]	19.43 [19.41, 19.45]
Ordered	21	4.96 [4.93, 4.99]	16.20 [16.13, 16.27]
SST-2	0	6.81 [6.61, 7.02]	24.53 [23.82, 25.29]
SST-2	14	3.44 [3.38, 3.50]	22.97 [22.67, 23.29]
SST-2	21	5.05 [4.90, 5.20]	25.40 [24.91, 25.91]

Fast–middle interaction grows from 3.33–9.20% at shift 16 to 16.20–26.94% at shift 128. The exact score-additivity proposition identifies these measured interactions with the mixed softmax response. The readouts quantify how the normalised attention map couples independent positional score components.

G.6 Separating positional and contextual changes under reordering

For the three alternative demonstration orders without a gap, we align tokens bijectively with the original prompt and construct four projected attention readouts o^{ab}. Index a selects the original or reordered positions; index b selects the corresponding observed de-rotated Q/K and V. Reapplying the selected rotations gives the four cells, with o^{00} and o^{11} reproducing the two observed endpoints. Define

$$\begin{aligned} &\Phi = o^{10} - o^{00}, \qquad C = o^{01} - o^{00}, \\ &J = o^{11} - o^{10} - o^{01} + o^{00}, \qquad T = o^{11} - o^{00} = \Phi + C + J. \end{aligned} \tag{G.3}$$

Here C includes changes in contextual Q/K/V produced by upstream computation. For component $X \in \{\Phi, C, J\}$, the pooled signed projection is $\gamma_X = \sum_r \langle X_r, T_r \rangle / \sum_r \|T_r\|_2^2$, where r ranges over the aligned reordered readouts. These directional contributions satisfy $\gamma_\Phi + \gamma_C + \gamma_J = 1$.

At layer 0, direct phase accounts for essentially the entire projected reordering response. At layers 14 and 21, the contextual-feature component has signed projection 0.935–0.976. The exact decomposition therefore resolves the transition from direct positional rotation to changes carried by the representations built through the network.

Table G.6: Signed projections of direct phase, contextual-feature, and interaction components onto the total projected change under actual reorderings. Each task/layer contains 256 held-out prompt sets.

Task	Layer	Phase γ_Φ	Context γ_C	Interaction γ_J
Corrupt	0	1.000	0.000	0.000
Corrupt	14	0.099	0.935	-0.034
Corrupt	21	0.028	0.964	0.009
Ordered	0	1.000	0.000	0.000
Ordered	14	-0.024	0.945	0.079
Ordered	21	-0.004	0.953	0.051
SST-2	0	1.000	0.000	0.000
SST-2	14	0.035	0.967	-0.002
SST-2	21	0.020	0.976	0.004

G.7 Attributing local cache deletions

We independently suppress each of the eight demonstration spans at a fixed attention readout, execute the resulting cache mask, and continue the model to measure its downstream margin change. For a deleted set G, its headwise attention mass $\alpha_G = \sum_{j\in G} p_j$ and weighted value $w_G = \sum_{j\in G} p_j v_j$ give the exact finite response

$$\Delta y_{-G} = \frac{\alpha_G y - w_G}{1 - \alpha_G}. \tag{G.4}$$

An immediate cache-management consequence is the checkable constraint $\|\sum_h \Delta y_{-G,h} W_{o,h}\|_2 \le \epsilon$: the chosen deletion respects a specified current-query write-distortion budget. General finite KV edits supply the corresponding criterion for quantization or restoration. This is a local decision rule; future queries require their own evaluation. The experiment below evaluates deletion-effect attribution. Projecting and summing heads, then contracting with the baseline downstream gradient, predicts the signed margin effect. One baseline backward pass supplies the gradient for all eight candidate spans. The Jacobian control uses $\alpha_G y - w_G$ before the same projection and gradient contraction. Attention mass and the norm of the projected weighted value supply magnitude ranking controls.

At layers 0 and 14, we rank the eight candidates by predicted effect magnitude and compute Spearman correlation with their measured absolute margin changes. Correlations are averaged over the five variants within a prompt, then over the 256 held-out prompts. Sign accuracy compares signed predictions with the executed local interventions.

Table G.7: Local cache-deletion attribution against executed downstream effects. The four central columns report magnitude-rank correlations; the final column reports finite-predictor sign accuracy. Each row uses 256 prompt sets.

Task	Layer	Finite	Jacobian	Attention	Value norm	Sign accuracy
Corrupt	0	0.989	0.980	0.570	0.542	99.49%
Corrupt	14	0.950	0.933	0.723	0.727	98.43%
Ordered	0	0.988	0.974	0.377	0.361	99.13%
Ordered	14	0.982	0.981	0.757	0.764	99.43%
SST-2	0	0.988	0.965	0.352	0.343	99.53%
SST-2	14	0.977	0.974	0.701	0.714	99.15%

Finite-direction attribution achieves rank correlations of 0.950–0.989 and 98.4–99.5% sign accuracy. The comparison connects the token-group decomposition to separately executed model responses and provides a local effect-ranking instrument alongside positional prediction.

G.8 Targeted precision repair

At layer 7, selected by lowest calibration mean relative repair RMSE, we apply simulated symmetric per-vector 2-bit or 4-bit quantisation to the prefix K and V. For a vector x and bit setting b, the scale is

$a = \max_k |x_k|/(2^{b-1} - 1)$ and the stored test vector is $a\ \mathrm{round}(x/a)$, with the zero vector handled separately. Quantised values are represented in FP32 for the native intervention. The current query's own key/value is retained at original precision. Repair is evaluated on the original-order, no-gap variant of each of the 256 test prompt sets per task.

For a budget $k \in \{1, 4\}$, exact greedy selection uses (D.23) after every restoration. The first-order and second-order controls expand (D.22) along the joint score/value restoration path, score candidates once in the initially corrupted state, and restore the top k entries. Attention mass, projected value norm, random selection, and a Fisher-style score $\sum_h p_{hj}(1 - p_{hj})\delta_{hj}^2$ provide additional controls. All methods have access to the original cache. Thus the multi-entry comparison evaluates exact greedy reselection against fixed-score selection; the one-entry comparison evaluates a single choice for both methods.

For prompt u and repaired combined write $\widehat{o}_u$, define

$$r_u = \sqrt{\frac{\|\widehat{o}_u - o_u\|_2^2}{\left\|o_u^{\mathrm{corrupt}} - o_u\right\|_2^2}}, \qquad \overline{r} = \frac{1}{256}\sum_{u=1}^{256} r_u. \tag{G.5}$$

Here o_u is the original projected write; unrepaired corruption has $r_u = 1$. Heads are projected and summed before squared error is measured. The reported quantity is the mean of prompt-level relative RMSEs.

Table G.8: Mean relative projected-write RMSE after restoring one or four prefix entries at layer 7. Each row averages 256 test prompt sets. Finite greedy recomputes benefits after each restoration; second-order scores are fixed at the initial corrupted state. Lower is better.

		One entry		Four entries	
Task	Bits	Finite	Second-order	Finite	Second-order
Corrupt	2	0.7806	0.7838	0.5874	0.7794
Corrupt	4	0.8807	0.9258	0.6173	0.8433
Ordered	2	0.7697	0.8378	0.5932	0.7477
Ordered	4	0.6811	0.6811	0.2870	0.6367
SST-2	2	0.7811	0.7815	0.5815	0.7835
SST-2	4	0.8073	0.8650	0.6079	0.7584

With one entry restored, both methods make a single choice from the same corrupted cache and physical-token candidates. Exact selection reduces mean relative RMSE by 4.86% on Corrupt at 4 bits, 8.13% on Ordered at 2 bits, and 6.67% on SST-2 at 4 bits. Their paired finite-minus-second-order intervals are $[-0.05475, -0.03639]$, $[-0.06828, -0.06791]$, and $[-0.07357, -0.04182]$, respectively. Across all six conditions, the finite point estimate is lower in five and equal on Ordered at 4 bits.

With four entries restored, exact greedy selection lowers mean relative RMSE by 19.8–54.9% against fixed second-order selection across the six task/precision conditions. All six paired difference intervals are below zero. The one-entry comparison measures candidate choice, while the four-entry comparison measures the complete repeated-restoration procedure. Because the re-scoring schedules differ, the four-entry result does not isolate the benefit of exact scoring from that of adaptive re-scoring.

G.9 Numerical validation

Native FP32 attention replay, physical KV edits, grouped-query layouts, batched intervention isolation, and downstream gradients are checked before checkpoint evaluation using tiny Qwen2 and Llama models. Float64 algebra checks and native FP32 replay checks use separate references; the saved absolute/relative tolerances are $2 \times 10^{-10}/2 \times 10^{-10}$ and $5 \times 10^{-5}/5 \times 10^{-4}$, respectively. The three completed Qwen task workers record 1,166,880 passing checks across their calibration and test stages, with no failed saved checks. Their model revision, prompt manifest, numerical settings, and shared calibration-policy hashes agree. Evaluation uses PyTorch 2.11.0+cu128 and Transformers 4.51.3; Corrupt and SST-2 run on A100 40GB and Ordered on an L4. The complete checkpoints retain predictions, measured effects, selection indices, per-prompt metrics, and named numerical checks.

The SmolLM2 extension uses an A100 80GB GPU, PyTorch `2.8.0.dev20250319+cu128`, and Transformers 4.51.3, with TF32 disabled. Additional checks on the pretrained checkpoint verify cached versus full-forward execution, installed no-op agreement, cache isolation, and a downstream gradient direction against a central finite difference. All 419,584 saved calibration and test checks pass under the existing algebraic and native tolerances.

H A RECORDED ANSWER-REPAIR CASE

The accompanying instrument records one synthetic API-expression repair on Qwen2.5-0.5B-Instruct, revision `7ae557604adf67be50417f59c2c2f167def9a775`, using CPU FP32 eager attention, PyTorch 2.8.0, and Transformers 4.57.3. Generation is greedy with a 32-token limit and independent caches. Generated expressions are mechanically qualified with the same API namespace in every branch before running the same six fixed executable checks.

The ranking objective is the next-token logit margin for `legacy` against `lookup`, supplied from the known API contract. The candidate family contains seven odds multipliers, deletion, value zeroing, and six band/shift combinations (± 16 in fast, middle, and slow bands). Scoring all token rows and layers gives 109,080 candidates from one captured baseline and one backward pass, with no candidate-specific model calls for scoring. The highest-ranked edit multiplies attention odds at prefix token 79, `␣legacy`, by 64 at layer 21 for the final query. The predicted margin change is 11.361 and its executed change is 9.399.

Table H.1: Recorded outcomes on the same six fixed checks. Ordinary generation and the attention branch use the same prompt. Source emphasis supplies the full current API source. The installed no-op reproduces ordinary output token for token. Expressions omit the mechanically added API namespace.

Branch	Generated expression	Checks passed
Ordinary	`lookup(value)`	0/6
Source emphasis	`current(value)`	1/6
Installed no-op	`lookup(value)`	0/6
Ranked attention edit	`legacy(current(value))`	6/6

The saved confirmation reports matching ordinary and attention prompt token IDs, an unchanged check, and reproduction of the passing edited branch. The case was selected adaptively and uses a supplied correct ranking target. It demonstrates an executed repair under these conditions; it supplies no success-rate estimate or claim that attention uniquely caused the failure. The figure-data file retains the source record hash and reported branch outcomes; full discovery, confirmation, and executable-case records belong with the instrument's supporting records.

I REFERENCE COMPARISON WITH AN INDEPENDENT REGRESSION LEARNER

This pilot compares trained prediction functions with the task-learning setup of von Oswald et al. (2023). It characterises the prediction functions selected by training relative to an independently specified regression-GD family. The exact gradient-step representation in (B.8) is established algebraically for arbitrary projection weights and holds irrespective of alignment with this reference.

Controlled training protocol Following the task-learning comparison of von Oswald et al. (2023), we train attention blocks on fresh linear-regression episodes. Each episode draws $w_* \sim \mathcal{N}(0, I_5/5)$ and Gaussian inputs $x_j \sim \mathcal{N}(0, I_5)$, with targets $t_j = x_j^\top w_*$. Sixteen support tokens $[x_j, t_j]$ form the context; an independent query is encoded as $[x, 0]$. The query reads the support bank and its prediction is the final coordinate of the residual output.

The three architectures are linear attention, softmax attention, and RoPE–softmax attention. Each uses one residual attention block with two heads of width 16 and learned query, key, value, and output projections. RoPE uses base 10,000. We train seeds $0, 1, 2$ for 2,000 steps with AdamW, learning rate 0.003, zero weight decay, batches of 128 fresh episodes, and gradient-norm clipping at 1. Architecture comparisons use matched training streams. Checkpoint selection minimises prediction

MSE on 128 validation episodes, evaluated every 200 training steps. Held-out prediction evaluation uses 256 test episodes with eight independent queries per episode.

A separately specified one-step reference For support set $D = \{(x_j, t_j)\}_{j=1}^N$, define

$$\mathcal{L}_D(w) = \frac{1}{2N}\sum_{j=1}^{N}(x_j^\top w - t_j)^2, \qquad g_D = \frac{1}{N}\sum_{j=1}^{N} x_j t_j, \qquad w_1 = 0 - \eta\nabla\mathcal{L}_D(0) = \eta g_D. \quad \text{(I.1)}$$

The resulting prediction is $f_{\eta,D}(x) = \eta x^\top g_D$, and its query Jacobian is $\nabla_x f_{\eta,D} = \eta g_D$. The regression rule is computed from support inputs and labels independently of the attention projections, scores, values, and effective matrices.

For each trained checkpoint, we calibrate one nonnegative scalar against its validation predictions $f_\theta(D, x)$:

$$\eta_{\text{align}} = \arg\min_{\eta\geq 0} \sum_{(D,x)\in\mathcal{V}} \left[f_\theta(D,x) - \eta x^\top g_D\right]^2. \quad \text{(I.2)}$$

This scalar is then fixed for test evaluation. Thus the reported comparison measures held-out agreement with a validation-aligned GD family. A constructed linear-attention positive control at $\eta = 0.7$ matches the reference prediction and query Jacobian to floating-point precision, with Jacobian cosine 1.0.

Held-out prediction and sensitivity alignment Table I.1 reports every training seed. Prediction disagreement is the squared difference between attention and the calibrated GD prediction, averaged over the full test bank and then the three seeds. Jacobian cosines are computed in float64 on sixteen held-out diagnostic episodes with eight queries each, using the gradient of the native scalar prediction with respect to the query's five input coordinates.

Table I.1: Courtesy regression comparison: trained attention versus the independently specified one-step GD family, with a validation-calibrated scale. The cosine columns show per-seed test means; prediction MSE is disagreement with the GD reference, averaged across all three seeds on the full test bank.

	Query-Jacobian cosine			Prediction
Architecture	Seed 0	Seed 1	Seed 2	disagreement MSE
Linear attention	0.9951	0.9945	0.9974	0.005946
Softmax attention	0.9628	0.9653	0.9632	0.026968
RoPE–softmax attention	0.8231	0.9639	0.9647	0.072063

The trained linear models attain cosine 0.994–0.997 with the regression-GD direction. The ranges are 0.963–0.965 for softmax and 0.823–0.965 for RoPE–softmax. The comparison describes functional alignment with an external learner and its seed variability; it does not identify the objective selected by training. The exact head representation is given by (B.8).

J Initial single-layer joint-KV experiment

This earlier experiment uses 114-token retrieval prompts at one model and layer. The supplemental sweep in Section F uses its own saved 112-token retrieval prompts and adds SST-2, another checkpoint size, and another 1.5B layer. The two runs are reported separately. Both use the predictors in (F.1)–(F.2) and the prompt-paired evaluation defined in Section F.

J.1 Paired prompts and executed cache interventions

We evaluate the interaction in (D.9) on frozen Qwen2.5-1.5B-Instruct (Qwen Team, 2024a)[1] at zero-indexed layer 21, with FP32 eager attention, 12 query heads, and two KV groups. The layer was fixed before this run using the positional-propagation measurements reported in Section G.

[1]Checkpoint revision: `989aa7980e4cf806f80c7fef2b1adb7bc71aa306`.

The dataset contains 32 paired entity–label retrieval prompts generated with seed 20260908. Each baseline lists eight records of the form `Entity 06: B`, followed by a question asking for one entity's A/B label. Its donor cyclically permutes entity identifiers and flips the label at every record position, retaining the question and format. All prompts contain 114 tokens including the chat template. Tokenisation verifies identical record-span positions at both endpoints. The eight record spans are the intervention candidates; every pair and every span is evaluated. Appendix M gives a complete paired example.

For each span, we execute three patches: donor keys only, donor values only, and donor keys and values together. Each replacement acts on all physical KV groups at layer 21. It uses the native rotated keys and native values captured from the donor prefix, while preserving the baseline query, self-token entry, and every unpatched cache entry. The remaining query computation is executed through the model. Throughout this experiment the target is the fixed margin $m = \mathrm{logit}(A) - \mathrm{logit}(B)$, and the measured effect is the patched margin minus the unpatched margin in the same counterfactual batch.

The complete design contains 256 joint patches and 512 key-only/value-only controls. Capture uses 64 prefix forward passes and 32 baseline backward passes. The 24 patches for each prompt share a counterfactual batch with one unpatched control, giving 768 patched and 32 control single-query continuations. Every batch row starts from an independent copy of the baseline cache.

J.2 Initial prediction and interaction results

Table J.1: Joint cache-intervention prediction on 32 paired prompts, eight spans per prompt, at Qwen2.5-1.5B-Instruct layer 21. The response column is contracted with the same baseline gradient after projecting and summing heads. MAE is in answer-logit-margin units; recall and sign accuracy are percentages. The zero predictor has undefined ranking and sign scores.

Predictor	Local response	MAE	Top-two recall	Sign accuracy
Zero	0	0.079689	—	—
AtP (score/value first order)	$L_s + V_0$	0.075484	64.06	81.25
AtP* (separate key/value)	$K_f + V_0$	0.081529	65.63	83.98
Quadratic interaction	$K_f + V_0 + C_2$	0.056043	68.75	85.16
Exact joint	$K_f + V_0 + C_{KV}$	**0.004075**	**96.88**	**99.22**
Dense equality control	$y' - y$	0.004075	96.88	99.22

Including C_{KV} reduces MAE by 95.0% against AtP*, by 94.6% against AtP, and by 92.7% against the quadratic interaction correction (Table J.1). The primary difference is -0.077455, with paired 95% interval $[-0.093958, -0.062308]$. The joint-minus-quadratic difference is -0.051969, with interval $[-0.065922, -0.039547]$. Every prompt pair improves against both comparisons. Figure J.1A displays the aggregate errors. The exact joint predictor recovers 62 of the 64 top-two span slots and predicts the sign of 254 of the 256 effects.

J.3 The added term tracks an independently executed interaction

The key-only and value-only controls isolate the two constituent responses. For these controls $C_{KV} = 0$, and the joint and AtP* predictors coincide: their downstream MAEs are 0.005767 for key-only patches and 0.002422 for value-only patches. Combining both edits introduces the interaction measured in Table J.1.

The executed margins supply the four-cell target in (F.4). Across the 256 spans, $\widehat{I}_m$ has Pearson correlation 0.9829 with I_m and MAE 0.007424. Figure J.1B compares interaction errors. The mean absolute executed interaction is 0.084483; the quadratic prediction $\mathcal{G}(C_2)$ has interaction MAE 0.059019. These measurements connect the improvement to the term identified algebraically: reallocating attention changes the effect of a simultaneous value edit. The instrument provides separate signed key, value, and interaction contributions for each candidate span.

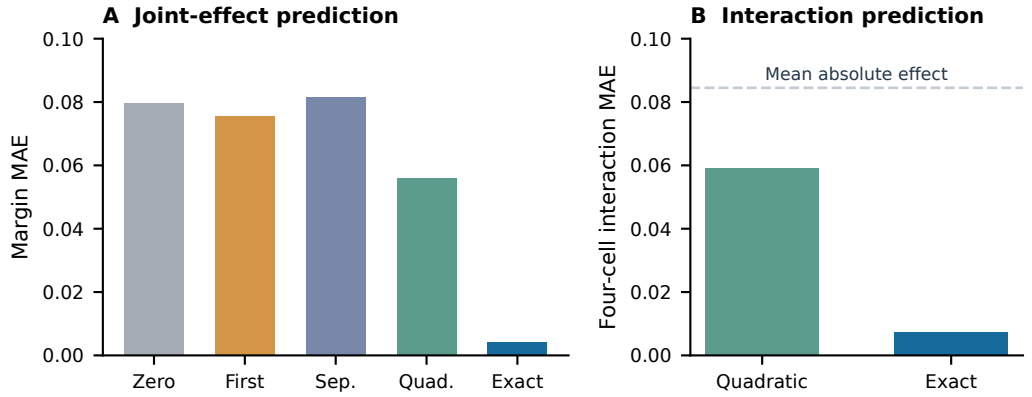

Figure J.1: Summary of the initial single-layer experiment. **A:** Reported mean downstream prediction errors (Table J.1); First and Sep. denote AtP and AtP*. **B:** Mean errors for predicting the executed four-cell interaction with $\mathcal{G}(C_2)$ and $\mathcal{G}(C_{KV})$; the dashed line is the mean absolute executed interaction. Bars are aggregate summaries over the 32 prompts and eight spans.

J.4 Numerical audit and reproducibility

All 9,504 recorded intervention checks and 1,729 preflight checks pass. Independent FP64 replay from the saved native tensors reproduces every joint prediction and interaction score within 3×10^{-15}. Against the executed FP32 projected write, the gradient-contracted local response differs by at most 2.6×10^{-6}. The largest change in the unpatched margin between single-query and counterfactual-batch execution is 4.6×10^{-5}; each measured intervention uses its own batch's unpatched control. Prompt texts, token spans, native Q/K/V, gradients, output projections, all predictors, executed margins, and per-prompt summaries are retained for offline replay. Appendix M specifies the reproducible run.

K SUPPLEMENTAL JOINT-KV PROTOCOL AND FULL RESULTS

K.1 CHECKPOINTS, PROMPTS, AND NATIVE EXECUTION

The supplemental run uses runner version 1.0.1, seed 20260908, PyTorch 2.11.0+cu128, Transformers 4.51.3, and CUDA 12.8 on an NVIDIA A100-SXM4-40GB. Execution uses FP32 eager attention in evaluation mode, frozen parameters, disabled TF32, and native RoPE at base 10^6. The 0.5B instruct checkpoint has 14 query heads, two KV heads, and head width 64; the 1.5B instruct checkpoint has 12 query heads, two KV heads, and width 128. The pinned checkpoint revisions are, respectively, `7ae557604adf67be50417f59c2c2f167def9a775` and `989aa7980e4cf806f80c7fef2b1adb7bc71aa306`.

The SST-2 snapshot is `stanfordnlp/sst2`, revision `8d51e7e4887a4caaa95b3fbebbf53c0490b58bbb`. Each query comes from the validation split and each of its eight demonstrations from the training split, balanced four per class. Query lengths are restricted to 20–320 characters and demonstration lengths to 20–160 characters before tokenisation; normalised duplicates within a prompt are excluded. The mapping is A for negative and B for positive. Retrieval uses the entity permutation and label-flip rule in Appendix M, with its own saved 112-token chat prompts. Both tokenizers validate all 64 prompt pairs before the sweep. The report archive includes the exact texts, source row IDs, token spans, and their hashes.

The final query reads a captured native prefix cache. The runner clones that cache for every counterfactual, patches native rotated keys and values, and executes the remaining query computation. A leaf at the attention output projection preserves the forward value while obtaining the baseline margin gradient with respect to that projected write. Every batch includes an unpatched control. Fractional edits save the actual rounded endpoint tensors used in both prediction and execution. Completed setting–pairs are committed atomically with hashes and the run fingerprint; the complete run records 192 such pairs.

The native checks compare full and cached continuation, unpatched controls, head readouts, projected writes, and gradient contractions. FP64 algebra and native FP32 comparisons use separate tolerances and references. The local algebra tests also reconstruct the affine-input gradient-step identity directly. The report archive contains all 5,120 scalar intervention records, 192 numerical audit rows, per-prompt summaries, the CPU/CUDA gate records, and the timing measurements. Native per-pair tensor captures are stored separately for offline replay; the report archive alone does not contain those tensors.

K.2 ALL FULL-REPLACEMENT PREDICTOR MEANS

Table K.1: Supplemental full-replacement downstream margin MAE. Each row evaluates 32 prompt pairs and eight joint interventions per pair. Model sizes denote Qwen2.5-Instruct checkpoints; layers are zero-indexed. Dense equality-control MAEs coincide with exact joint to the displayed precision.

Model	Task	Layer	Zero	AtP	AtP*	Quadratic	Exact joint
0.5B	Retrieval	18	0.001521	0.001028	0.001042	2.98×10^{-4}	4.37×10^{-6}
0.5B	SST-2	18	0.007884	0.003183	0.003099	5.99×10^{-4}	2.64×10^{-5}
1.5B	Retrieval	14	0.008275	0.003796	0.003342	0.001525	8.22×10^{-5}
1.5B	Retrieval	21	0.098962	0.112774	0.107864	0.087683	0.007844
1.5B	SST-2	14	0.012270	0.004656	0.003194	0.001418	3.42×10^{-4}
1.5B	SST-2	21	0.108527	0.089914	0.109616	0.080700	0.004510

Table K.2: Prompt-paired exact-minus-comparator MAE differences at full replacement. Negative values favor exact joint. Intervals are individual 95% percentile intervals from 5,000 resamples of the 32 prompt pairs (seed 981), retaining all eight spans. All 12 displayed intervals and the 12 additional intervals against zero and AtP lie below zero.

Setting (model / task / layer)	Δ_{AtP^*}	95% interval	Δ_{quad}	95% interval
0.5B / Retrieval / 18	-0.001037	$[-0.001148, -0.000935]$	-0.000293	$[-0.000352, -0.000241]$
0.5B / SST-2 / 18	-0.003072	$[-0.003433, -0.002720]$	-0.000572	$[-0.000657, -0.000487]$
1.5B / Retrieval / 14	-0.003260	$[-0.003859, -0.002716]$	-0.001442	$[-0.001800, -0.001126]$
1.5B / Retrieval / 21	-0.100020	$[-0.129003, -0.076014]$	-0.079840	$[-0.105230, -0.056382]$
1.5B / SST-2 / 14	-0.002852	$[-0.003714, -0.002079]$	-0.001076	$[-0.001371, -0.000791]$
1.5B / SST-2 / 21	-0.105106	$[-0.120056, -0.090357]$	-0.076190	$[-0.091271, -0.060908]$

Table K.3: Prediction of the independently executed four-cell margin interaction I_m. The exact and quadratic predictions are $\mathcal{G}(C_{KV})$ and $\mathcal{G}(C_2)$. Correlations use signed effects and are descriptive; each row contains 256 spans clustered within 32 prompt pairs. The final column supplies the absolute scale of the executed interaction.

Setting (model / task / layer)	Exact MAE	Quadratic MAE	Exact r	Mean $\lvert I_m \rvert$
0.5B / Retrieval / 18	4.52×10^{-6}	2.96×10^{-4}	0.99999	0.001043
0.5B / SST-2 / 18	2.54×10^{-5}	5.95×10^{-4}	0.99998	0.003094
1.5B / Retrieval / 14	6.00×10^{-5}	0.001513	0.99979	0.003321
1.5B / Retrieval / 21	0.008826	0.092810	0.99698	0.115820
1.5B / SST-2 / 14	1.62×10^{-4}	0.001331	0.99837	0.003308
1.5B / SST-2 / 21	0.003902	0.076860	0.99857	0.110007

K.3 Second-model validation on SmolLM2

We repeat the matched joint-KV protocol on `HuggingFaceTB/SmolLM2-1.7B-Instruct`, revision `31b70e2e869a7173562077fd711b654946d38674`, at layers 12 and 18. The run uses an NVIDIA A100-SXM4-80GB, PyTorch 2.8.0+cu128, Transformers 4.51.3, and FP32 eager attention. The raw retrieval and SST-2 prompts match the Qwen run before tokenisation; SmolLM2 retrieval prompts have 117 tokens and SST-2 prompts have 213–316 tokens. The four settings contain 32 prompt pairs and eight spans each. Full-strength key-only, value-only, and joint replacements contribute 3,072 interventions; joint edits at $\alpha = 0.1$ and 0.5 for retrieval layer 18 add 512, giving 3,584 unique intervention records.

Table K.4: Prompt-paired SmolLM2 exact-minus-comparator MAE differences at full replacement. Negative values favour exact joint. Intervals are individual 95% percentile intervals from 5,000 resamples of the 32 prompt pairs, retaining all eight spans.

Setting (task / layer)	Δ_{AtP^*}	95% interval	Δ_{quad}	95% interval
Retrieval / 12	-0.0006587	$[-0.0008378, -0.0005199]$	-0.0004522	$[-0.0005475, -0.0003720]$
Retrieval / 18	-0.0319843	$[-0.0371174, -0.0273633]$	-0.0244537	$[-0.0296274, -0.0194620]$
SST-2 / 12	-0.0030330	$[-0.0041765, -0.0020356]$	-0.0021496	$[-0.0029157, -0.0014187]$
SST-2 / 18	-0.0880370	$[-0.1013916, -0.0761594]$	-0.0371866	$[-0.0467584, -0.0287562]$

Across these four settings, exact joint reduces MAE by 83.8–99.3% relative to AtP* and by 78.6–99.0% relative to the quadratic correction. All eight paired intervals in Table K.4 lie below zero. The correlation between the exact local interaction prediction and the independently executed downstream four-cell interaction ranges from 0.99447 to 0.99997.

All 128 saved prompt-pair captures reproduce their numerical-audit fields in offline replay. The maximum FP64 sparse–dense local discrepancy is 7.76×10^{-15}, the maximum discrepancy from the native FP32 local contraction is 1.46×10^{-6}, the maximum full-forward versus cached difference is 1.91×10^{-5}, and the maximum unpatched control-margin drift is 5.72×10^{-6}. These checks validate the local implementation; the downstream prediction still contracts the exact local response with one baseline gradient and is not itself an algebraic identity.

Table K.5: SmolLM2 retrieval layer-18 strength diagnostic. Difference intervals are prompt-paired exact-minus-quadratic 95% intervals.

α	Quadratic MAE	Exact joint MAE	Difference 95% CI
0.1	0.00003266	0.000006149	$[-0.00003451, -0.00001930]$
0.5	0.003406	0.00007076	$[-0.004110, -0.002593]$
1.0	0.024695	0.0002412	$[-0.029627, -0.019462]$

K.4 Complete local runtime comparison

All candidates are batched, with no automatic chunk fallback or precision change. Synthetic cases use native-style Q/K/V tensors and eight-token spans; the native capture has eight seven-token spans. PyTorch benchmark timing uses warmup, accelerator synchronisation, five randomised rounds, and a minimum run time of 0.12 seconds per measurement. Totals directly measure the preparation and evaluation specified in Section F.6; component medians need not sum to total medians. All seven timing audits pass against dense and FP64 references.

Table K.6: Directly measured local evaluation time including preparation (ms). $R = 1$ evaluates one strength; $R = 3$ shares preparation across three strengths. Ratios are dense time divided by sparse time, so a ratio below one denotes a sparse slowdown. Values are medians of five timing rounds on one A100; component timings, interquartile ranges, and raw measurements are archived.

	$R = 1$ (ms)		$R = 3$ (ms)		Dense / sparse	
Case	Dense	Sparse	Dense	Sparse	$R = 1$	$R = 3$
Native: $N = 112, C = 8$	2.083	2.521	4.014	3.292	0.83	1.22
Synthetic: $N = 128, C = 8$	2.096	2.517	4.021	3.296	0.83	1.22
Synthetic: $N = 128, C = 256$	2.133	2.533	4.146	3.279	0.84	1.26
Synthetic: $N = 1024, C = 8$	2.087	2.516	4.024	3.279	0.83	1.23
Synthetic: $N = 1024, C = 256$	2.134	2.537	4.157	3.288	0.84	1.26
Synthetic: $N = 8192, C = 8$	2.068	2.498	3.979	3.256	0.83	1.22
Synthetic: $N = 8192, C = 256$	2.142	2.531	4.197	3.281	0.85	1.28

L COMPLETE FREQUENCY-BAND PREDICTION MEANS

Table L.1 reports the Qwen band-by-shift strata for all three tasks. Each row averages 3,840 executed interventions from 256 prompt sets, using layers 0, 14, and 21 and all five prompt variants. The corresponding SmolLM2 strata are reported in Table L.2; each row averages 2,560 interventions from 256 prompt sets, using layers 12 and 18 and five variants. The zero-predictor MAE gives the mean absolute measured margin change.

Bands are defined separately from each checkpoint's rotary frequencies. For SmolLM2, the maximum fast-, middle-, and slow-band angles at shift 16 are 16, 0.2794, and 0.004878 radians; at shift 128 they are 128, 2.2351, and 0.039028 radians. These differ from the Qwen scales, so equal token displacements do not imply equal angular displacements across models.

Table L.1: Downstream positional-prediction MAE by frequency band and displacement, in logit-margin units. All rows use the same finite, Jacobian, and zero predictors.

Task	Band	Shift	Finite MAE	Jacobian MAE	Zero MAE
Corrupt	Fast	16	0.001026	0.066144	0.024164
Corrupt	Fast	128	0.108527	0.567056	0.166353
Corrupt	Middle	16	0.000064	0.000236	0.004576
Corrupt	Middle	128	0.003543	0.017862	0.031212
Corrupt	Slow	16	0.000006	0.000006	0.000119
Corrupt	Slow	128	0.000006	0.000009	0.000953
Corrupt	All	16	0.001391	0.066527	0.026921
Corrupt	All	128	0.110114	0.564376	0.203602
Ordered	Fast	16	0.089449	0.174306	0.140993
Ordered	Fast	128	0.131197	0.681604	0.177739
Ordered	Middle	16	0.000057	0.000402	0.007160
Ordered	Middle	128	0.002239	0.027925	0.038921
Ordered	Slow	16	0.000004	0.000004	0.000222
Ordered	Slow	128	0.000005	0.000014	0.001787
Ordered	All	16	0.080270	0.164970	0.138236
Ordered	All	128	0.159647	0.702497	0.220769
SST-2	Fast	16	0.002061	0.056467	0.022879
SST-2	Fast	128	0.143155	0.557336	0.218531
SST-2	Middle	16	0.000040	0.000241	0.004042
SST-2	Middle	128	0.002112	0.016428	0.027545
SST-2	Slow	16	0.000006	0.000006	0.000169
SST-2	Slow	128	0.000006	0.000010	0.001353
SST-2	All	16	0.002239	0.056709	0.024750
SST-2	All	128	0.137185	0.546868	0.235175

Table L.2: SmolLM2 downstream positional-prediction MAE by frequency band and displacement, in logit-margin units. Each row pools layers 12 and 18 and five variants within each of 256 prompt sets.

Task	Band	Shift	Finite MAE	Jacobian MAE	Zero MAE
Corrupt	Fast	16	0.000402	0.082632	0.019618
Corrupt	Fast	128	0.001455	0.632637	0.037069
Corrupt	Middle	16	0.000127	0.001282	0.008357
Corrupt	Middle	128	0.005623	0.053762	0.061269
Corrupt	Slow	16	0.000008	0.000011	0.001359
Corrupt	Slow	128	0.000334	0.000639	0.011332
Corrupt	All	16	0.000659	0.082800	0.022440
Corrupt	All	128	0.017234	0.636586	0.063771
Ordered	Fast	16	0.001366	0.107675	0.014865
Ordered	Fast	128	0.004256	0.878299	0.095722
Ordered	Middle	16	0.000326	0.003567	0.016982
Ordered	Middle	128	0.005255	0.141881	0.056700
Ordered	Slow	16	0.000030	0.000094	0.006645
Ordered	Slow	128	0.002210	0.006434	0.058842
Ordered	All	16	0.002332	0.104332	0.029464
Ordered	All	128	0.043074	0.877506	0.156422
SST-2	Fast	16	0.000373	0.056496	0.019089
SST-2	Fast	128	0.001146	0.447842	0.030308
SST-2	Middle	16	0.000094	0.001229	0.009065
SST-2	Middle	128	0.005209	0.064702	0.061861
SST-2	Slow	16	0.000010	0.000014	0.001630
SST-2	Slow	128	0.000495	0.000884	0.013632
SST-2	All	16	0.000505	0.056739	0.020938
SST-2	All	128	0.011178	0.453083	0.080708

M JOINT-KV PROTOCOL AND REPRODUCTION

The initial joint experiment in Appendix J fixes one model, one layer, one dataset family, and one primary comparison before execution. For pair index $u \in \{0, \ldots, 31\}$, NumPy's random generator is initialised with `SeedSequence([20260908, u])`. It samples eight A/B labels, ensures both labels occur, shuffles them, and selects the query entity. The donor rotates the entity-identifier array by a sampled nonzero cyclic shift and flips each row's label. The complete prompts and token spans are saved before evaluation. Every record includes its terminating newline in the patched span; the query's own cache entry remains in the attended bank.

Table M.1: The first paired prompt. Both endpoints use the instruction *Read the records. Answer the question with A or B only.* and finish with *Question: What is the label of Entity 06?* followed by *Answer:*. The native Qwen chat template wraps the instruction, records, and question.

Record position	Baseline	Donor
1	`Entity 01: A`	`Entity 04: B`
2	`Entity 02: B`	`Entity 05: A`
3	`Entity 03: A`	`Entity 06: B`
4	`Entity 04: B`	`Entity 07: A`
5	`Entity 05: B`	`Entity 08: A`
6	`Entity 06: B`	`Entity 01: A`
7	`Entity 07: B`	`Entity 02: A`
8	`Entity 08: A`	`Entity 03: B`

For each record position the donor supplies the native keys and values at the same token indices. All predictors receive identical captures. The fixed-value key correction evaluates the complete span jointly, retaining its common softmax denominator. The value correction uses baseline attention weights; the joint correction uses the edited attention weights. For multiple heads, the output

projection and head sum precede the downstream-gradient contraction. The complete per-intervention table reports both the signed C_{KV} correction and the executed factorial effect, together with the residual $I_m - \mathcal{G}(C_{KV})$.

The run uses Transformers 4.51.3, PyTorch 2.11.0+cu128, and an NVIDIA A100 SXM4 40GB. Native execution is FP32 with TF32 disabled; local algebra and effective-matrix audits use FP64. The effective readout audit uses the mean-key score-shift convention, $k_j - \overline{k}$, and includes each endpoint's uniform value mean. The baseline statistics and positive complement mass implement the sparse formula in (D.15).

The standalone command is

```
python rope_joint_kv_experiment.py --device cuda \
    --batch 32 --out joint_kv_results
```

The supplied `Joint_KV_Attribution_Colab.ipynb` embeds the same source. Native tiny-Qwen preflights check cache isolation, grouped-query attention, batch parity, full-forward parity, and the baseline gradient before the pretrained checkpoint is downloaded. Each completed prompt pair is committed with file hashes and the protocol fingerprint, enabling interrupted runs to resume from the next pair.

The primary interval resamples prompt pairs with replacement 5,000 times using bootstrap seed 981. The quadratic contrast uses the same sampling scheme. The release tables retain all 768 interventions, the 32 prompt-level summaries, and the model/source manifest. The figure-generation script reads these tables directly. Native tensor captures additionally retain Q/K/V, the output projection, the common gradient, batched controls, and the executed outputs so each predictor can be reconstructed without rerunning the model.

N CONTROLLED NUMERIC READOUTS AND STATE-DEPENDENT PREDICTION

The experiments in Section 5.5 separate exact observation-level operations from the prediction advantage of state-dependent weights. Both use scalar observed values and an affine final readout. They do not require propagation through a remaining nonlinear network.

N.1 NUMERIC READOUT, DATA, AND TRAINING

The first model has one head of width $d = 16$, a constant pre-RoPE key $k = (1, \dots, 1, 0, \dots, 0)$ with eight entries in each half, and frequencies $\omega_r = 10000^{-r/8}$ for $r = 0, \dots, 7$. Its values are the sixteen lagged observations. The observed state $z \in \{-1, 1\}$ is balanced and independent of the time series. With the query $q(z) = q_0 + zq_1$, its probabilities are

$$p_j(z) = \mathrm{softmax}_j\left(\frac{q(z)R(-j)k^\top}{\sqrt{16}}\right), \qquad j = 1, \dots, 16.$$

The forecast is (19). It contains no normalisation layer, contextual encoder, residual branch, or feed-forward block. At fixed state, $w_j(z) = \beta p_j(z)$ is the exact coefficient on observation x_j and its signed contribution is $w_j(z)x_j$. For $\bar{y} = \sum_j p_j x_j$, singleton masking satisfies

$$\Delta\widehat{y}_{-j} = \beta\frac{p_j(\bar{y} - x_j)}{1 - p_j}. \tag{N.1}$$

The implementation uses the complement mean to evaluate this response when $1 - p_j$ is poorly resolved. Masking does not refit the model or close a gap in the observation positions. Group masking uses its joint normaliser, not the sum of (N.1) over the group.

The generator is

$$y_t = 0.45y_{t-1} + 0.15y_{t-4} + 0.10y_{t-16} + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0, 0.3^2). \tag{N.2}$$

Each of seeds $13, 37, 73, 101, 149$ generates separate trajectories with 512 burn-in observations for training, validation, and test, retaining 4,096, 1,024, and 1,024 observations respectively. Query coefficients are initialised independently from $\mathcal{N}(0, 0.03^2)$; the output gain starts at 0.5 and the intercept at zero. Adam uses learning rate 0.025, moments $(0.9, 0.999)$, denominator offset 10^{-8}, and 1,000 updates with 256 training rows sampled with replacement per update. Validation MSE is evaluated every twenty updates and selects the checkpoint. OLS fits an intercept, all sixteen lags, and the same state on the training data. The test trajectories are not used for selection.

N.2 FIXED INTERVENTIONS AND NUMERICAL CHECKS

Eight evenly spaced test contexts per seed are selected without using their forecast errors. Each context receives sixteen edit templates: deletion; reweighting by $1/4, 1, 4$; value scaling by $-1/2$; positional shifts $16, 128$ in each of the fast, middle, slow, and full bands; and key, value, or joint replacement from the final cache entry. Bands partition the frequency-ordered planes into thirds. Every singleton is edited, together with the fixed groups of zero-indexed rows $\{0, 2\}$ and $\{0, 1, 2, 3\}$. A replacement can be a no-op when it uses its own donor; those cases are retained.

Two additional fixtures use generator seed 51031. The first has four query heads, two KV groups, three query rows, nine keys, and causal endpoints $3, 6, 8$. Its head width is eight and its value width is three. The second has five scalar values and a dominant score of 45 against four zero scores. Frequencies in both fixtures are $(1, 0.1, 0.01, 0.001)$. Interventions that empty an attended row are excluded by the specified masking rule.

The oracle forms complete edited keys and values, applies RoPE through complex multiplication, and recomputes full softmax and readouts. It does not call the finite-response implementation. Table N.1 reports the checks by operation; their multiplicities are numerical comparisons, not independent statistical samples. Elementwise acceptance uses absolute and relative tolerances 2×10^{-9}. Training gradients are checked by centred differences with step 10^{-5}. Both the exact finite-score softmax tangent and the positional Jacobian are retained as approximation controls. The joint response records K_f, V_0, C_{KV} separately; the rotary audit records the nonadditive output interaction between fast and middle bands.

Table N.1: Numerical checks for the observation-level experiment. The complete run contains 47,066 passing comparisons over 11,807 finite interventions, together with reconstruction, gradient, lag, and timing-pipeline checks.

Check	Comparisons	Maximum absolute error
Training gradient	25	8.64×10^{-13}
Final forecast reconstruction	40	1.11×10^{-16}
Independent dense readout	42	2.22×10^{-16}
Observation-weight reconstruction	42	2.22×10^{-16}
Contribution sum	42	8.88×10^{-16}
Effective-step reconstruction	49	5.55×10^{-16}
Finite response output	11,807	1.11×10^{-15}
Instrument contraction	11,807	1.92×10^{-15}
Sparse response	11,807	1.82×10^{-15}
Vectorised singleton response	10,464	2.11×10^{-15}
Singleton or group masking	737	1.67×10^{-15}
$K_f + V_0 + C_{KV}$ sum	42	1.33×10^{-15}
Common positional translation	42	2.22×10^{-16}
Lag-rotation composition	42	7.11×10^{-15}
Disjoint-band score additivity	42	1.78×10^{-15}
Vectorised scenario construction	12	3.55×10^{-15}
Prepared sparse–dense agreement	12	5.55×10^{-16}
Complete-pipeline agreement	12	5.55×10^{-16}

N.3 Preparation and scenario reuse

The scenario sweep uses the last trained numeric checkpoint, state $z = 1$, and independent Gaussian values from generator seed 7401. Context lengths are $64, 256, 1024$; candidate counts are $1, 8, 64, 256$. Every candidate shifts the same four most recent entries in the full rotary band, with displacements evenly spaced from -128 to 128; the singleton grid uses -128. The length sweep measures computation, not extrapolation accuracy.

A prepared complement stores its log mass and projected value mean. Each candidate then evaluates only its four edited entries and combines them with that complement in log space. This implements (17); the ordinary `Scorer.local` path in the instrument still recomputes full softmax. The dense comparator batches candidates in chunks of at most 64, also reusing baseline logits. Scenario construction is vectorised and shared by both methods. Every timed candidate is checked against this comparator.

Table N.2 retains the complete grid. Each timing is the median of seven measurements after a warmup. Prepared evaluation excludes baseline preparation, complement preparation, and score-increment construction. The preparation-inclusive measurements directly time the complete local pipelines, including those operations where required; they are not sums of separately timed components. The first experiment is timed without the second running concurrently. Both use CPU float64 with NumPy 2.3.5 and Python 3.12.14 on x86-64 Linux. These measurements exclude training, model loading, and any pretrained-model forward or backward pass.

N.4 Proof of the two-lag separation

This proves Theorem 2 in Section 3.4.

Proof. The residual $U = Z(X_1 - X_2)/2$ is orthogonal in L^2 to each of $1, X_1, X_2, Z$, and $\mathbb{E}[U^2] = 1/2$. Hence the projection of Y onto $\mathcal{L}$ is $(X_1 + X_2)/2$. Independence of the noise gives (13).

For the RoPE construction, use row vectors and put $r_j = kR(-\ell_j)^\top$ at the two lags. By assumption $a = r_1 - r_2 \neq 0$. Choose

$$q_\kappa(Z) = \frac{\tau\kappa Z}{\|a\|_2^2} a, \qquad \tau = \sqrt{d}.$$

Table N.2: Complete CPU scenario grid for the numeric readout. Ratios are dense time divided by sparse time; values below one favour dense evaluation. Prepared ratios exclude setup. Millisecond timings include baseline preparation, any required group preparation, and common edit construction.

		Prepared	Including preparation		
N	Candidates	Ratio	Dense (ms)	Sparse (ms)	Ratio
64	1	1.33	0.0988	0.1186	0.83
64	8	1.42	0.1278	0.1442	0.89
64	64	1.77	0.1711	0.1891	0.90
64	256	4.13	0.5978	0.2903	2.06
256	1	1.31	0.1122	0.1660	0.68
256	8	1.58	0.1624	0.1556	1.04
256	64	2.95	0.3545	0.4610	0.77
256	256	8.46	0.7207	0.3140	2.30
1,024	1	1.44	0.1386	0.1642	0.84
1,024	8	2.44	0.1867	0.2271	0.82
1,024	64	22.02	0.9443	0.2300	4.11
1,024	256	35.00	2.4039	0.3597	6.68

The score difference is $q_\kappa(Z)(r_1 - r_2)^\top/\tau = \kappa Z$, so two-entry softmax gives the displayed weights. Conditional on either state, the noise-free prediction error is, up to sign, $[1 - \mathrm{sigmoid}(\kappa)](X_1 - X_2)$. Its second moment is $2/(1 + e^\kappa)^2$; independent noise adds σ_ε^2. Finally, $\mathrm{sigmoid}(\kappa Z) = 1/2 + Z\tanh(\kappa/2)/2$ gives (15). □

N.5 Two-lag training, comparators, and intervals

The second experiment uses independent episodes, not overlapping windows from one series. For each seed, the training, validation, and test sets contain 4,096, 1,024, and 4,096 episodes. Each draws $X_1, X_2 \sim \mathcal{N}(0, 1)$, a balanced independent Z, and $\varepsilon \sim \mathcal{N}(0, 0.1^2)$. The switching and linear-control targets share those inputs and noise. The seed list is $13, 37, 73, 101, 149, 211, 269, 331, 401, 463$.

The head has width two, rotary frequency one, constant key $(1, 0)$, and lags one and two. Its query follows the construction in Section N.4, leaving only κ trainable. Starting at $\kappa = 0$, Adam uses learning rate 0.08, moments $(0.9, 0.999)$, denominator offset 10^{-8}, and 800 full-batch updates. Validation MSE is evaluated every ten updates and selects the checkpoint. The recorded switching-task checkpoints have $\kappa \in [4.4015, 4.5296]$.

Raw-input OLS fits the features $(1, X_1, X_2, Z)$. The expanded comparator adds ZX_1, ZX_2. Removing state dependence and refitting attention gives a constant gate $cX_1 + (1 - c)X_2$, with

$$c = \mathrm{clip}_{[0,1]} \frac{\sum_i (X_{i1} - X_{i2})(Y_i - X_{i2})}{\sum_i (X_{i1} - X_{i2})^2},$$

computed only on the training set. A uniform gate is also retained. Tables N.4–N.6 give every seed for both experiments.

The paired interval resamples complete independent data and training seeds with replacement 10,000 times, using bootstrap seed 70031. The statistical unit is the complete replicate, not an individual intervention or test episode. The recorded switching criteria are a positive lower 95% endpoint for the OLS-minus-RoPE MSE difference, a mean reduction of at least 0.1, and at least 0.9 attention on the selected lag in both states for every replicate. All three pass. The linear and interaction-feature controls are retained independently of those criteria.

The first, second, and final test episodes for each task and seed receive the finite-intervention audit. Independent dense RoPE execution and the instrument agree with the trained gate, and (15) is checked over every test episode. All 8,380 numerical checks pass. The experiment demonstrates learned state-dependent weighting on the specified inputs; it does not test learning an unobserved regime or compare general nonlinear forecasting architectures.

Table N.3: Aggregate prediction with and without state-dependent weights. Entries are mean held-out MSE over ten independent seeds. Constant refits a single attention weight; interactions adds ZX_1, ZX_2 to OLS.

Task	RoPE	Raw OLS	Constant	Interactions
Switching	0.010193	0.516066	0.515848	0.009953
Linear control	0.009949	0.009944	0.009931	0.009953

Table N.4: Every autoregressive training seed. Entries are test MSE; the generator's noise variance is 0.09.

Seed	RoPE	OLS on lags and state	Training mean
13	0.097465	0.094370	0.126824
37	0.098718	0.096244	0.137945
73	0.095308	0.091725	0.128520
101	0.100760	0.097892	0.139418
149	0.097679	0.094453	0.134241

Table N.5: Every switching-task seed. Entries are test MSE; κ is the validation-selected gate parameter.

Seed	κ	RoPE	Raw OLS	Constant	Interactions
13	4.4809	0.010435	0.503126	0.503301	0.010239
37	4.4105	0.009743	0.523299	0.522920	0.009468
73	4.4015	0.010330	0.499801	0.498944	0.010060
101	4.4474	0.010291	0.528063	0.527996	0.009965
149	4.4917	0.010199	0.518517	0.518261	0.010026
211	4.4529	0.010024	0.512882	0.511516	0.009829
269	4.5103	0.010368	0.520066	0.520396	0.010171
331	4.5296	0.010067	0.535589	0.536337	0.009833
401	4.4850	0.010385	0.518986	0.518937	0.010085
463	4.4649	0.010087	0.500334	0.499876	0.009853

Table N.6: Every linear-control seed. Entries are test MSE. Constant and interactions use the same definitions as Table N.3.

Seed	RoPE	Raw OLS	Constant	Interactions
13	0.010212	0.010215	0.010205	0.010239
37	0.009454	0.009456	0.009451	0.009468
73	0.010056	0.010055	0.010054	0.010060
101	0.010015	0.009962	0.009935	0.009965
149	0.010026	0.010024	0.010016	0.010026
211	0.009801	0.009812	0.009809	0.009829
269	0.010156	0.010158	0.010140	0.010171
331	0.009808	0.009816	0.009801	0.009833
401	0.010067	0.010092	0.010058	0.010085
463	0.009897	0.009847	0.009839	0.009853

Reproduction and scope. Both experiment scripts are supplied as separate research artifacts, with the unchanged instrument, complete scenario tables, training traces, fitted parameters, per-seed metrics, and source hashes. The commands are

```
python experiment_transparency.py --profile paper --out runs/transparency
python experiment_more.py --profile paper --out runs/more
```

The second artifact also retains all training, validation, and test episodes. The transparency artifact retains its test trajectories and checkpoints. These runs use controlled numeric heads, with no new pretrained Qwen or GPU execution. In a deep-model capture, the same identities concern cached hidden values and a fixed-gradient local proxy. Exact final-forecast claims here rely on the affine output in (19).