A WRONG TURN IN ADDITION: GENERATION, ROUTING, AND CORRECTION OF ARITHMETIC ERRORS IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite rapid progress in mathematical reasoning, large language models remain surprisingly brittle on elementary arithmetic. We ask how a local arithmetic error becomes a persistent autoregressive trajectory, and whether evidence of that error survives after the model has committed to it. Using matched digit-preserved carry counterfactuals and causal interventions, we decompose multi-operand addition errors into generation, read-back routing, and downstream compatibility. Natural off-by-one carry errors are characterized by weakened carry-sensitive influence, while the emitted digit subsequently selects between local lower- and higher-carry continuation routes through a localized query–key interaction. Crucially, error propagation and error representation coexist: the model can continue along a wrong route while downstream states remain sensitive to whether the read-back digit is arithmetically compatible with its context. This delayed compatibility signal is visible in next-token behavior, entropy, and representation geometry, and enables a training-free mechanism-guided correction method that substantially improves exact-answer accuracy and outperforms most training-required baselines.

1 INTRODUCTION

Frontier reasoning models now achieve gold-medal-level performance at the International Mathematical Olympiad and have begun solving research-level open mathematical problems (Luong et al., 2025; Feng et al., 2026). Yet exact arithmetic remains unexpectedly brittle: even frontier models can fail on ordinary multi-digit addition, with accuracy degrading as operands grow longer and many failures traceable to carry handling or digit alignment (Bie et al., 2026; Yan et al., 2025). This contrast makes addition a useful microscope for studying computation inside LLMs: the algorithm is elementary and its intermediate variables are explicit, but the model must realize them through distributed autoregressive representations rather than an external calculator.

Recent work has begun to uncover how LLMs perform arithmetic internally. Mechanistic studies have reverse-engineered arithmetic algorithms in small Transformers (Nanda et al., 2023; Quirke & Barez, 2024), while analyses of pretrained LLMs recover structured numerical and error-related representations (Zhou et al., 2024; Sun et al., 2025; Wen et al., 2026). Carry handling has also been linked to limited lookahead in multi-operand addition (Baeumel et al., 2025) and to geometric structure underlying modulo-10 off-by-one digit errors (Wen et al., 2026). These findings establish carry processing as a central source of arithmetic failure, but leave its causal dynamics unresolved: *how does a carry mistake arise at the current position, and how does the emitted wrong digit subsequently alter the computation of the next one?*

Figure 1 summarizes the perspective we take. An addition error unfolds in two stages. At the target position p, the model selects among nearby digit outcomes associated with different carry possibilities and may choose the wrong one. Once generated, that digit is fed back into the model—or *read back*—as part of the prefix for position $p + 1$, where it can redirect the computation of the next digit. We trace the first wrong digit in a naturally generated answer and ask three questions: why the wrong carry-related alternative is selected at p; how feeding that digit back changes the computation at $p + 1$; and whether later states still distinguish the resulting continuation from one

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.

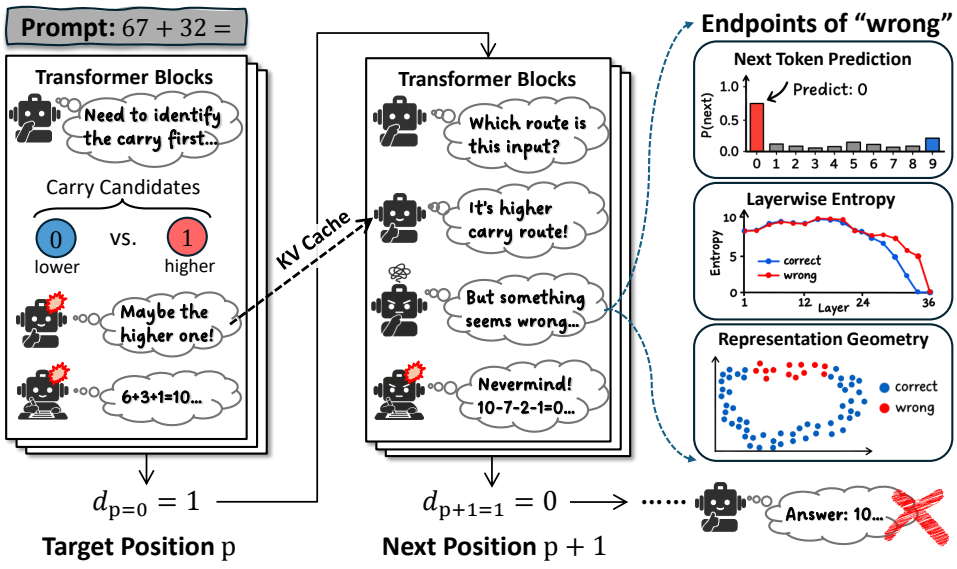

Figure 1: **Illustration of a two-stage wrong turn in addition.** (**Left**) When asked $67+32 = 99$, the model must first resolve competing carry hypotheses when predicting the answer. At target position $p = 0$, the model evaluates competing carry hypotheses. An erroneous preference for the higher carry yields the wrong digit $d_p = 1$. (**Middle**) Once emitted and read back, this digit acts as a query that interacts with the cached target position key. This QK routing locks the subsequent computation into an incorrect carry-conditioned path $d_{p+1} = 0$. (**Right**) Although the model continues this wrong route, its arithmetic incompatibility with the true context remains visible in downstream endpoints, reflected in skewed next-token predictions, elevated late-layer entropy, and shifted geometry of the final-layer hidden states. Details see Appendix C.

consistent with the arithmetic. We answer these questions using matched counterfactuals that vary the carry-producing context or the fed-back digit while preserving the relevant answer trajectory.

Using a Qwen3 model as the primary system, we find a concrete mechanism across these two stages. At the first stage, carry-related computation has a weaker effect on the target-digit decision in natural error cases than in correct cases, and intervening on this computation can reverse or induce the corresponding digit choice. At the second stage, once the selected digit is read back, two late attention heads route it toward locally lower- or higher-carry continuations, with localized $Q_{p+1} \cdot K_p$ interactions accounting for most of its effect on the next prediction. Thus, a carry mistake is not merely an incorrect output token; after read-back, it redirects the continuation along which subsequent arithmetic proceeds[1].

Crucially, following a carry-conditioned route does not erase whether the read-back digit fits the preceding arithmetic. In our matched cases, the same digit produces different downstream states when it is wrong in the original context versus correct in the counterfactual context. This difference appears in next-token margins, late-layer entropy, and representation geometry. Thus, a wrong digit can still redirect the next computation, while later states retain evidence that it did not fit the arithmetic—a delayed signal that can be used for correction.

Finally, we derive a training-free inference-time correction method building upon the identified mechanism. The lower- and higher-carry routes propose two candidates, whose next-pass uncertainty determines which branch to continue. This method substantially improves answer accuracy and outperforms most training-required baselines.

In summary, the main contributions of this work are:

- We identify weakened carry-sensitive influence at the target position and uncover how the emitted digit subsequently routes continuation through lower-/higher-carry QK interactions in Transformers and recurrent writes in Gated DeltaNet models.

[1]Across additional Transformer and Gated DeltaNet models, this same computational recurs through architecture-specific mechanisms (see Appendix D).

- We show that route commitment does not erase arithmetic structure: downstream states retain compatibility-sensitive signals in next-token behavior, late-layer entropy, and representation geometry.
- We translate these mechanisms into a training-free inference-time correction method that substantially improves answer accuracy and outperforms most training-required baselines.

2 Preliminaries and Data Construction

2.1 Autoregressive Multi-Operand Addition

Generally, we consider the addition of n integers $A_0, \ldots, A_{n-1}$. Let $a_{i,k} \in \{0, \ldots, 9\}$ denote the digit of A_i at decimal significance 10^k, with $k = 0$ corresponding to the units place. If the generated answer contains P digit positions, we index them from left to right by $p \in \{0, \ldots, P-1\}$, with $p = 0$ denoting the first generated digit. Position p corresponds to decimal significance $k = P - 1 - p$.

At position p, we define the raw column sum (position-wise sum) and ground-truth digit as

$$r_p = \sum_{i=0}^{n-1} a_{i,P-1-p}, \qquad d_p^{\mathrm{GT}} = (r_p + c_p) \pmod{10}, \tag{1}$$

where c_p is the incoming carry propagated from less significant positions. Our main experiments employ Qwen3-4B-Instruct-2507 (Yang et al., 2025) on the addition of three ten-digit integers (Figure 2), for which $c_p \in \{0, 1, 2\}$. We focus on answer position $p = 1$ in the main text, where natural first errors are sufficiently frequent for controlled mechanistic analysis.[2]

We write $h_p^{(L)}$ for the layer-L residual state used to predict d_p. Once d_p is emitted, it is appended to the prefix and read back by the model, producing $h_{p+1}^{(L)}$ for the prediction of d_{p+1}. We therefore distinguish the *target position* p, where the current digit is generated, from the *next position* $p + 1$, where that digit has become part of the input context for subsequent inference. A *natural first error* at p has a correct generated prefix $d_{<p} = d_{<p}^{\mathrm{GT}}$ but an incorrect prediction $d_p = d_p^W \neq d_p^{\mathrm{GT}}$, where the superscript W means *wrong*.

Here, we focus on *off-by-one errors*, the dominant class of addition errors. We define them by $d_p^W = d_p^{\mathrm{GT}} \pm 1$[3], where 0 and 9 are also one step apart. Following Wen et al. (2026), we let $c_p^W \in \{0, 1, 2\}$ denote the alternative carry hypothesis satisfying $d_p^W = (r_p + c_p^W) \pmod{10}$. We call $c_p^W = c_p + 1$ an *over-carry* error and $c_p^W = c_p - 1$ an *under-carry* error.

2.2 Matched Counterfactual Setup

Starting from an original problem X with an off-by-one first error as defined above, we construct three matched cases as illustrated in Figure 2. Case $\mathcal{A}$ is the original trajectory, where the model naturally emits d_p^W. Case $\mathcal{B}$ keeps X unchanged but teacher-forces the ground-truth digit d_p^{GT}, changing only the digit that is read back at the next position $p + 1$.

For Case $\mathcal{C}$, we construct a new problem X' by modifying lower-significance operands so that the carry into position p changes while the target raw sum is preserved. We require

$$d_p'^{\mathrm{GT}} = d_p^W = d_p^{\mathrm{GT}} \pm 1 \pmod{10}, \qquad d_j'^{\mathrm{GT}} = d_j^{\mathrm{GT}} \ (j \neq p), \qquad d'_{\leq p} = d'^{\mathrm{GT}}_{\leq p}. \tag{2}$$

Thus, the original wrong digit becomes the correct digit of X', all other ground-truth answer digits are unchanged, and the model naturally generates the correct trajectory through position p on X'.

The two contrasts serve different purposes. $\mathcal{A} \leftrightarrow \mathcal{B}$ asks how changing only the read-back digit affects the next computation while keeping the arithmetic problem fixed. $\mathcal{A} \leftrightarrow \mathcal{C}$ asks whether the same read-back digit is processed differently when the surrounding arithmetic context changes so that the digit becomes correct rather than incorrect. Thus, $\mathcal{A}/\mathcal{B}$ tests the effect of digit identity, whereas $\mathcal{A}/\mathcal{C}$ tests the effect of context. All three cases share the same ground-truth continuation after position p, allowing their downstream behavior to be compared directly.

[2] Further evaluations on other positions and models are included in Appendices B and D.

[3] Digit arithmetic is modulo 10 throughout this paper unless stated otherwise.

2.3 Dataset Construction

Let $\mathcal{T} = \{0 \to 1,\ 1 \to 0,\ 1 \to 2,\ 2 \to 1\}$ denote the four admissible off-by-one carry transitions, where $a \to b$ means that the ground-truth incoming carry is $c_p = a$ while the alternative digit corresponds to carry hypothesis b.

For each analyzed position p, we construct two balanced datasets. The *first-error dataset*

$$\mathcal{D}_p^{\mathrm{FE}} = \bigcup_{(a\to b)\in\mathcal{T}} \mathcal{D}_{p,a\to b}^{\mathrm{FE}}$$

contains 200 natural first-error problems X^{FE} per transition (800 in total), with $c_p = a$ and $c_p^W = b$. For each X^{FE}, we construct its digit-preserved carry-flip problem X'^{FE} as in Section 2.2, yielding Cases $\mathcal{A}/\mathcal{B}/\mathcal{C}$.

The *correct-control dataset*

$$\mathcal{D}_p^{\mathrm{CC}} = \bigcup_{(a\to b)\in\mathcal{T}} \mathcal{D}_{p,a\to b}^{\mathrm{CC}}$$

contains 200 matched pairs $(X^{\mathrm{CC}}, X'^{\mathrm{CC}})$ per transition, where the model naturally generates both problems correctly through position p. The pair undergoes the same carry-flip construction, changing the incoming carry from $c_p = a$ in X^{CC} to $c'_p = b$ in X'^{CC}. We sample 200 pairs for $0 \to 1$ and $2 \to 1$, and split the 400 $c_p = 1$ controls evenly into 200 $1 \to 0$ and 200 $1 \to 2$ pairs. Details appear in Appendix A.

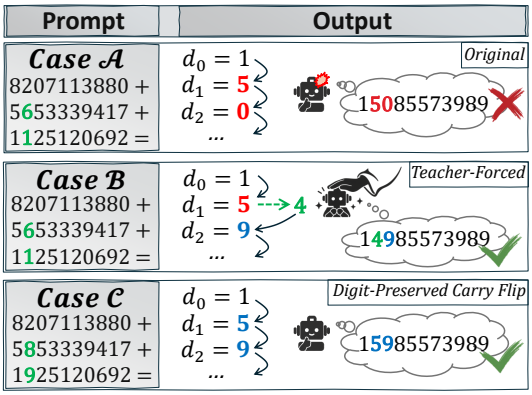

Figure 2: **Matched Case $\mathcal{A}/\mathcal{B}/\mathcal{C}$ construction.** $\mathcal{A}$ is the natural first-error trajectory. $\mathcal{A} \leftrightarrow \mathcal{B}$ holds the prompt fixed and changes only the read-back target digit by teacher-forcing the ground truth, whereas $\mathcal{A} \leftrightarrow \mathcal{C}$ preserves the read-back digit and changes its arithmetic context through a digit-preserved carry flip. All three cases share the preceding answer prefix and the same next ground-truth digit. Green marks externally imposed edits, while red and blue denote model-generated tokens.

3 How Errors Are Generated

Natural first errors in addition are overwhelmingly off-by-one: in our screened cohort at $p = 1$, 254 of 278 errors (91.37%) differ from the ground-truth digit by one (see Appendix A.2). What computation fails when the model makes such an error? As a one-step change in incoming carry induces exactly this shift at fixed raw sum r_p, we first examine carry-sensitive computation at the target position.

Counterfactual activation patching. We first ask how strongly a carry-changing counterfactual can alter the target-digit decision. Throughout the paper, $\mathcal{X} \leftarrow \mathcal{Y}$ denotes activation patching from donor branch $\mathcal{Y}$ into recipient branch $\mathcal{X}$, followed by downstream recomputation.

Let $z_q(d)$ denote the pre-softmax logit of digit d at position q. At the target position, we define

$$\mathrm{Margin}_p = z_p(d_p'^{\mathrm{GT}}) - z_p(d_p^{\mathrm{GT}}), \qquad \Delta\mathrm{Margin}_p^{\mathcal{X}\leftarrow\mathcal{Y}} = \mathrm{Margin}_p^{\mathcal{X}\leftarrow\mathcal{Y}} - \mathrm{Margin}_p^{\mathcal{X}}. \tag{3}$$

Thus, $\Delta\mathrm{Margin}_p$ measures the intervention-induced shift toward the carry-flipped answer.

We apply the scan to both $\mathcal{D}_p^{\mathrm{FE}}$ and $\mathcal{D}_p^{\mathrm{CC}}$. For first-error examples, we patch $\mathcal{A} \leftarrow \mathcal{C}$; for correct controls, we analogously patch $X^{\mathrm{CC}} \leftarrow X'^{\mathrm{CC}}$. In both cases, the donor is the matched carry-flipped counterpart of the original recipient. As Figure 3 (a) shows, errors exhibit a substantially weaker $\Delta\mathrm{Margin}_p$ than correct controls. Scanning attention and FFN outputs across layers further localizes the strongest carry-sensitive effect to L24 attention. Thus, natural first errors are distinguished not by the absence of a carry-sensitive pathway, but by its weaker influence on the target-digit decision.

Causal rescue and induction. We next test whether the carry-sensitive directions can causally control the target-digit decision. For each tested component, the steering direction is computed from the matched pair at that same component and then injected back into the original branch. For a first-error example, x^{FE} and x'^{FE} denote the activations of Case $\mathcal{A}$ and its matched Case $\mathcal{C}$; rescue

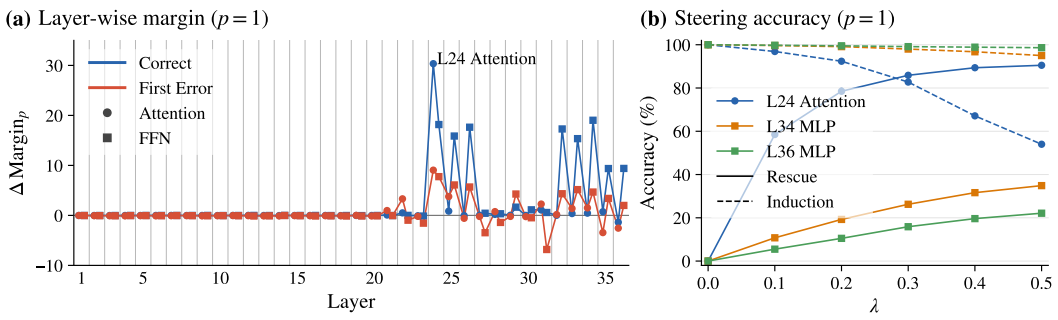

Figure 3: **Carry-sensitive control of the target decision at $p = 1$.** (**a**) Carry-counterfactual patching elicits a weaker margin response on natural first errors than on correct controls, with the strongest effect at L24 attention. (**b**) L24 attention also shows the strongest bidirectional steering effect, rescuing errors and inducing the corresponding errors on correct controls.

modifies the Case $\mathcal{A}$ activation away from the carry-flipped state. For a correct-control pair, x^{CC} and x'^{CC} denote the activations on X^{CC} and X'^{CC}; induction modifies the original X^{CC} activation toward its matched one-step carry-flipped counterpart:

$$x^{\mathrm{FE}}_{\mathrm{rescue}} = x^{\mathrm{FE}} + \lambda(x^{\mathrm{FE}} - x'^{\mathrm{FE}}), \qquad x^{\mathrm{CC}}_{\mathrm{induction}} = x^{\mathrm{CC}} + \lambda(x'^{\mathrm{CC}} - x^{\mathrm{CC}}). \tag{4}$$

After replacing the selected activation, we recompute the remaining forward pass and measure the original ground-truth digit accuracy. Figure 3 (b) applies this intervention independently to L24 attention, L34 FFN, and L36 FFN. L24 attention shows the strongest bidirectional control: rescue raises accuracy on $\mathcal{D}_p^{\mathrm{FE}}$ from 0% to 90.5%, while induction lowers accuracy on $\mathcal{D}_p^{\mathrm{CC}}$ from 100% to 54.0% at $\lambda = 0.5$. In contrast, steering the later L34/L36 FFNs yields substantially weaker rescue and little induction. Together, these results identify L24 attention as the strongest tested causal site through which the carry-sensitive direction controls the target decision.

4 Errors Become Routes

We next ask what happens after the wrong digit is emitted and read back. We show that the read-back digit does not merely perturb the next prediction, but selects between two carry-conditioned continuation routes.

Read-back digit controls the next prediction. We first ask whether the emitted target digit actively shapes the computation at the next position. Across 800 examples in $\mathcal{D}^{\mathrm{FE}}_{p=1}$, Case $\mathcal{A}$ predicts the next digit correctly in only 41.5% of cases (468 incorrect samples), whereas reading back the ground-truth digit in Case $\mathcal{B}$ raises accuracy to 98.38% (Table 1). Strikingly, replacing d_p^W with an off-route digit sampled randomly from the eight digits excluding $\{d_p^W, d_p^{\mathrm{GT}}\}$ still yields 89.63% next-digit accuracy. Thus, the failure in Case $\mathcal{A}$ is not simply a consequence of reading back an incorrect token. Instead, d_p^W and d_p^{GT} appear to occupy privileged, structured continuation routes, motivating the hypothesis that they correspond to competing local carry hypotheses.

Table 1: **Read-back effect at $p = 1$.** Off-route digits are unsupported by either routing head.

Read-back	Next Acc.	Recovered
Natural wrong ($\mathcal{A}$)	41.5%	–
GT digit ($\mathcal{B}$)	98.38%	457/468
Off-route digit	89.63%	397/468

Localizing the read-back pathway. We next localize how the read-back digit affects the computation at $p + 1$. Since $\mathcal{A}$ and $\mathcal{B}$ share the same d_{p+1}^{GT}, we define the next-position margin as

$$\mathrm{Margin}_{p+1} = z_{p+1}(d_{p+1}^{\mathrm{GT}}) - \max_{d \in \{0,\ldots,9\},\, d \neq d_{p+1}^{\mathrm{GT}}} z_{p+1}(d). \tag{5}$$

A larger Margin_{p+1} means stronger preference for the correct next digit, whereas a smaller or negative margin indicates a weaker or incorrect continuation. We use the same patched-minus-recipient convention as in Eq. (3) to define $\Delta\mathrm{Margin}_{p+1}^{\mathcal{X} \leftarrow \mathcal{Y}}$. We scan both $\mathcal{A} \leftarrow \mathcal{B}$ and $\mathcal{B} \leftarrow \mathcal{A}$ directions on development triplets. The layerwise scan reveals a strong early read-in effect across

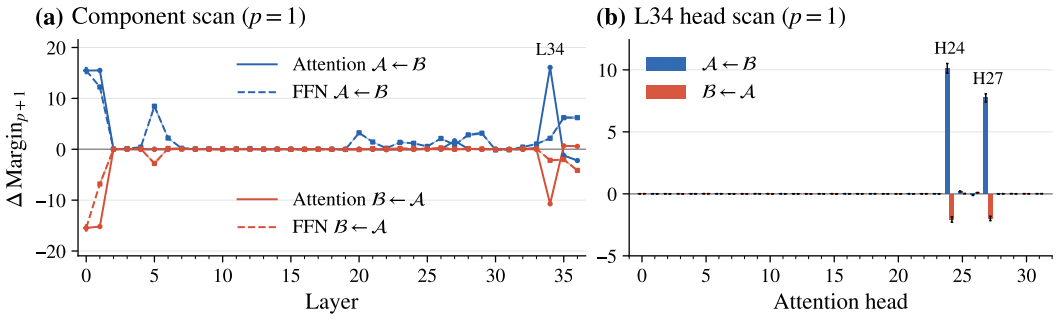

Figure 4: **Localization of read-back routing at $p = 1$.** Bidirectional activation patching through **(a)** component scan and **(b)** L34 head scan localizes the read-back routing pathway to L34H24 and L34H27.

the first two layers and a distinct late attention effect at L34 (Figure 4). Head-wise pattern patching at L34 further localizes the effect to H24 and H27, motivating the target-edge analysis below[4].

Causal closure at the target $Q{\cdot}K$ edge. We next test whether the L34H24/L34H27 routing effect is concentrated on the attention edge from $p+1$ to the cached p state. For query head $h \in \{\text{L34H24}, \text{L34H27}\}$, define its target-edge score

$$\ell_h = \frac{\langle Q_{p+1,h}, K_{p,h} \rangle}{\sqrt{d_{\text{head}}}}. \tag{6}$$

Table 2: **Target-edge causal closure at $p = 1$.**

Intervention	$\Delta\text{Margin}_{p+1}^{\mathcal{A}\leftarrow\mathcal{B}}$
H24 edge	+9.74 logits
H27 edge	+7.58 logits
H24+H27 edges	+16.04 logits
H24+H27 full	+16.15 logits
L34 all-head full	+16.07 logits

We compare five interventions of $\mathcal{A}\leftarrow\mathcal{B}$: replacing only ℓ_{H24}, only ℓ_{H27}, both scores jointly, the complete attention patterns of H24 and H27, or the full attention patterns of all L34 heads. For the edge interventions, all other attention logits are left unchanged before the softmax and downstream recomputation. We report the resulting mean $\Delta\text{Margin}_{p+1}^{\mathcal{A}\leftarrow\mathcal{B}}$ defined above. As Table 2 shows, patching both target edges nearly matches both full H24/H27 pattern replacement and full L34 attention replacement, indicating that almost all of the L34 routing effect is concentrated on these two $Q_{p+1}{\cdot}K_p$ interactions.

H24 and H27 route lower and higher carry hypotheses. We now ask which read-back digits preferentially activate H24 and H27. For each example, we hold the preceding arithmetic context fixed and separately read back the three candidate digits $d \in \{d_p{-}1, d_p, d_p{+}1\}$. For each candidate, we recompute the next-position query and measure its target-edge score

$$\ell_h(d) = \frac{\langle Q_{p+1,h}(d), K_{p,h} \rangle}{\sqrt{d_{\text{head}}}}, \qquad d_h^\star = \arg\max_{d\in\{d_p-1,d_p,d_p+1\}} \ell_h(d). \tag{7}$$

Thus, $d_h^\star$ is the read-back digit that most strongly activates head h toward the cached p state.

We plot this head assignment against the carry potential Φ_p introduced by Wen et al. (2026) $\Phi_p = \sum_{j=1}^{P-1-p} \frac{r_{p+j}}{10^j}$, whose integer boundaries determine the incoming carry $c_p = \lfloor \Phi_p \rfloor$. Figure 5 therefore shows, for each supplied digit, which of H24 or H27 treats that digit as its locally preferred candidate under the current arithmetic context.

As shown, most errors cluster near the boundaries at $\Phi_p = 1$ and $\Phi_p = 2$, consistent with the concentration of off-by-one errors near carry transitions observed in Wen et al. (2026). The joint distribution of correct controls and Case $\mathcal{A}$ gives the two heads a clear arithmetic interpretation. Immediately above an integer boundary i, the true carry is $c_p = i$ and the competing under-carry hypothesis is $i - 1$. Correct controls therefore predominantly select H27, whereas Case $\mathcal{A}$ under-carry errors select H24. Immediately below the same boundary, the true carry is $i - 1$ and the competing over-carry hypothesis is i; the assignments reverse, with correct controls on H24 and Case $\mathcal{A}$ errors on H27. Thus, H24 and H27 represent the lower and higher endpoints of the locally relevant carry pair, rather than fixed absolute carry values or fixed wrong/correct roles.

[4]Notably, these two heads share the same KV group in Qwen3-4B-Instruct-2507, suggesting that they access a common cached context through distinct query-side routing.

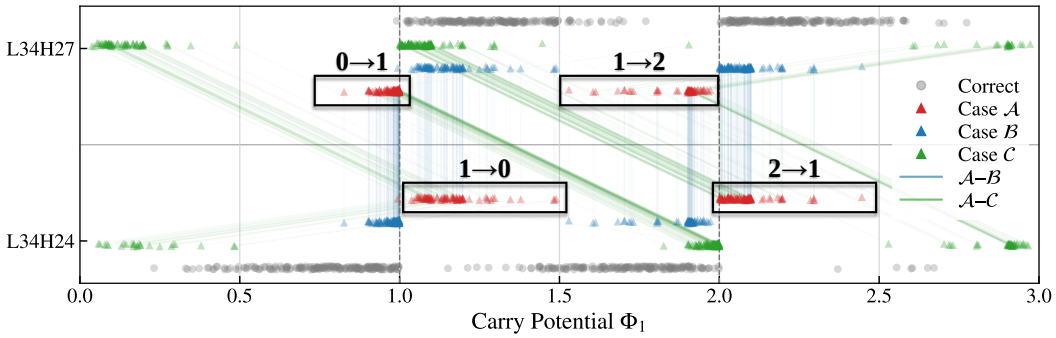

Figure 5: **Lower- and higher-carry routing at $p = 1$.** The horizontal axis shows carry potential Φ_1. The vertical axis records the unique head whose local candidate argmax equals the supplied read-back digit. Gray circles denote correct controls, while red, blue, and green triangles denote Cases $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$. Blue and green lines connect matched $\mathcal{A}/\mathcal{B}$ and $\mathcal{A}/\mathcal{C}$ pairs, respectively. Integer boundaries mark changes in the GT incoming carry. See Appendix B for additional analysis.

The matched connections further support this interpretation. Cases $\mathcal{A}$ and $\mathcal{B}$ share the same problem and therefore the same Φ_p (so their blue connectors are vertical). Replacing the wrong read-back digit with d_p^{GT} consistently moves the assignment to the opposite carry head. Cases $\mathcal{A}$ and $\mathcal{C}$ instead preserve the read-back digit while flipping the carry-producing context, shifting Φ_p across an adjacent integer boundary (producing diagonal green connectors). Most $\mathcal{A}/\mathcal{C}$ pairs also switch heads despite sharing the same read-back digit, showing that H24/H27 encode context-relative lower/higher carry roles rather than fixed digit identities.

Together, these results show how an emitted digit becomes a continuation route: after read-back, its query selects between locally lower- and higher-carry alternatives encoded relative to the preceding arithmetic context. The same organization recurs in other model architectures (see Appendix D).

5 Arithmetic Compatibility

The routing mechanism above explains how a local carry mistake becomes a persistent continuation, but not what arithmetic information remains after route commitment. Since even small Transformers can implement exact multi-digit addition circuits (Quirke & Barez, 2024; Cho et al., 2025), a local routing error need not erase arithmetic structure downstream. Indeed, a probe at $p + 1$ distinguishes whether the read-back digit d_p is correct more accurately than the analogous probe at p (Appendix C.4). We therefore ask whether downstream states still register arithmetic incompatibility after committing to a carry-conditioned route.

Wrong routes remain distinguishable downstream. Figure 6 shows that the incompatible branch $\mathcal{A}$ remains clearly separated from both compatible controls after read-back: $\mathcal{B}$ replaces d_p^W with d_p^{GT}, while $\mathcal{C}$ keeps the same digit but makes it compatible by changing the context. Using the same Margin_{p+1} defined in Eq. (5), $\mathcal{A}$ has a markedly lower next-position margin, markedly higher late-layer predictive entropy, and a larger geometric deviation than both $\mathcal{B}$ and $\mathcal{C}$. Thus, even after committing to a wrong route, the incompatible branch remains distinguishable from matched compatible branches in downstream endpoints.

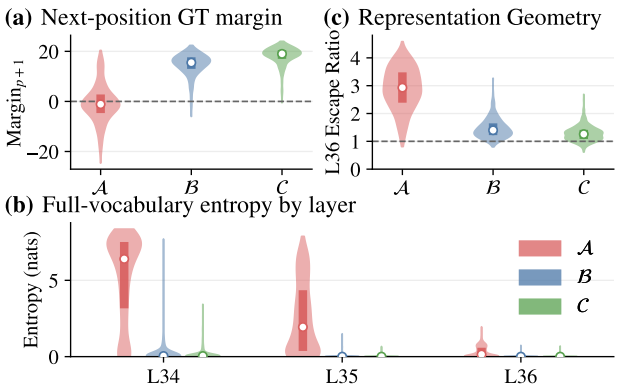

Figure 6: **Downstream read-back endpoints at $p = 1$.** Distributions over matched $\mathcal{A}/\mathcal{B}/\mathcal{C}$ branches: (**a**) next-digit GT margin, (**b**) entropy at L34–L36, and (**c**) L36 5-NN escape ratio (see Appendix C.3) relative to correct states. Dashed lines mark zero margin and unit escape ratio.

Digit and context meet at the routing edge. We next find that this compatibility-sensitive distinction is also expressed through the same $Q_{p+1} \cdot K_p$ interaction that implements carry routing. For each branch $\mathcal{X}$, let $Q^{\mathcal{X}}_{p+1,h}$ and $K^{\mathcal{X}}_{p,h}$ denote the query and cached key in Eq. (6) for $h \in \{\mathrm{L34H24}, \mathrm{L34H27}\}$. We denote $K^{\mathcal{A}}_p = K^{\mathcal{B}}_p = K^{\mathcal{A}/\mathcal{B}}_p$ as $\mathcal{A}$ and $\mathcal{B}$ share the same arithmetic context through position p. The three queries are elicited by the read-back digits d^W_p, d^{GT}_p, and d^W_p in Cases $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, respectively. We form all six distinct query–key pairings and insert their H24+H27 target-edge scores into the same Case $\mathcal{A}$ recipient, leaving the remaining attention computation unchanged.

The resulting pattern is strongly context dependent (Table 3). Under the original-context key $K^{\mathcal{A}/\mathcal{B}}_p$, only $Q^{\mathcal{B}}_{p+1}$ produces a large positive margin, while $Q^{\mathcal{A}}_{p+1}$ and $Q^{\mathcal{C}}_{p+1}$ remain near the incompatible baseline. Under the carry-flipped key $K^{\mathcal{C}}_p$, the pattern reverses: the two d^W_p-conditioned queries $Q^{\mathcal{A}}_{p+1}$ and $Q^{\mathcal{C}}_{p+1}$ produce the strongest margins. The remaining pairing, $Q^{\mathcal{B}}_{p+1}K^{\mathcal{C}}_p$, is weaker but need not become erroneous because d^{GT}_p is not guaranteed to belong to either carry route under the counterfactual context. When it is off-route, the behavior is consistent with the off-route control in Table 1.

Table 3: **Query–key recombination at $p = 1$.** Next-position margins for all L34 H24/H27 query–key pairings.

Margin_{p+1}	$K^{\mathcal{A}/\mathcal{B}}_p$	$K^{\mathcal{C}}_p$
$Q^{\mathcal{A}}_{p+1}$	-0.73	$+14.34$
$Q^{\mathcal{B}}_{p+1}$	$+15.41$	$+9.72$
$Q^{\mathcal{C}}_{p+1}$	-0.52	$+14.49$

Thus, the compatibility effect is relational rather than attached to a fixed query or key. The cached K_p provides the arithmetic-context-conditioned side of the local carry alternatives, while Q_{p+1} acts as a probe conditioned on the digit just read back. Their $Q_{p+1} \cdot K_p$ interaction determines how strongly that digit engages the locally available carry routes. When the selected route is incompatible with the context, the resulting branch remains distinguishable downstream in Margin_{p+1}, predictive entropy, and representation geometry (Figure 6).

As this signal emerges only after a candidate digit has been read back, standard autoregressive decoding cannot use it to revise the already emitted digit. It can, however, provide good guidance for comparison among tentative candidates, motivating our correction method in Section 6.

6 Mechanism-Guided Inference-Time Correction

The preceding analysis provides two complementary signals for correction. H24/H27 identify the locally relevant carry alternative, while entropy across L34–L36 strongly separates compatible from incompatible read-back states (Figure 6). We combine them into a training-free inference-time rule requiring neither labeled correction data nor a learned verifier.

Route-guided entropy correction. Let d_p be the model's greedy proposal. We compare its H24/H27 target-edge preferences to determine whether d_p can be uniquely assigned to the lower- or higher-carry route. When such an assignment is available, it identifies a single adjacent alternative. Otherwise, we fall back to the full three-candidate neighborhood:

$$\mathcal{C}_p(d_p) = \begin{cases} \{d_p, d_p + 1\}, & d_p = d^\star_{\mathrm{H24}} \neq d^\star_{\mathrm{H27}}, \\ \{d_p, d_p - 1\}, & d_p = d^\star_{\mathrm{H27}} \neq d^\star_{\mathrm{H24}}, \\ \{d_p - 1, d_p, d_p + 1\}, & \text{otherwise.} \end{cases} \tag{8}$$

For each $d \in \mathcal{C}_p(d_p)$, we tentatively read it back and compute the next-position full-vocabulary entropy $H^{(L)}_{p+1}(d)$ at $L \in \mathcal{L} = \{\mathrm{L34}, \mathrm{L35}, \mathrm{L36}\}$. We then commit a replacement only when all three layers select the same minimum-entropy candidate[5]:

$$\widehat{d}^{(L)} = \arg\min_{d \in \mathcal{C}_p(d_p)} H^{(L)}_{p+1}(d), \qquad \widetilde{d}_p = \begin{cases} \widehat{d}^{(L34)}, & \widehat{d}^{(L34)} = \widehat{d}^{(L35)} = \widehat{d}^{(L36)}, \\ d_p, & \text{otherwise.} \end{cases} \tag{9}$$

Thus, routing proposes the competing carry hypothesis and downstream uncertainty decides whether to accept it. We retain d_p when the layers disagree.

[5]Although L34 alone is highly discriminative, single-layer entropy selection has relatively lower false-positive preservation. Detailed ablations appear in Appendix E.4.

Baselines and results. We evaluate this method against several training-required correction baselines, including re-prompting (Sun et al., 2025), linear steering (Bhalla et al., 2024), hard replacement with a Target-state probe, and the Dual-stream probe of Wen et al. (2026). We also introduce another training-required Next-state probe that predicts the target digit after read-back, exploiting the stronger separability available at $p + 1$.

Route+Entropy reaches 96.57% token accuracy and 81.7% question (whole answer) accuracy, outperforming most training-required baselines and approaching the Next-state probe at 84.4%. The latter benefits from labeled supervision, whereas Route+Entropy uses only routing and uncertainty signals already present in the model.

Entropy is a second training-free variant that applies the same L34–L36 consensus to all three local candidates $\{d_p - 1, d_p, d_p + 1\}$. Its lower question accuracy of 77.9% shows that retaining the off-route alternative degrades correction. This provides a validation of the routing mechanism beyond its causal characterization above. Additional analyses and results for other models appear in Appendix E.

Table 4: **Inference-time correction.** Test on 1,000 held-out questions.

Method	Token Acc.	Ques Acc.
Original	92.07%	60.9%
Training-required		
Re-prompting	92.17%	61.4%
Steering	95.50%	63.5%
Dual-stream probe	93.71%	59.3%
Target-state probe	96.09%	66.4%
Next-state probe (ours)	**97.96%**	**84.4%**
Training-free (ours)		
Entropy	95.82%	77.9%
Route+Entropy	**96.57%**	**81.7%**

7 Related Work

Arithmetic mechanisms and errors. Mechanistic studies have recovered digit-wise addition algorithms in task-trained Transformers (Quirke & Barez, 2024; Quirke et al., 2025), while analyses of pretrained LLMs identify arithmetic computation through heuristic circuits and structured numerical representations (Nikankin et al., 2025; Zhou et al., 2024). For multi-operand addition, carry handling is a central failure mode: limited lookahead can break under cascading carries (Baeumel et al., 2025), while Wen et al. (2026) relate off-by-one digit errors to the geometry and quantization of carry representations. We build on this work by following a natural first error across time, from the carry-sensitive computation that produces it to the routing induced after read-back.

Internal error signals and correction. Internal states can retain information about errors even when generation is incorrect. Arithmetic probes can decode predicted and ground-truth answers and support selective correction (Sun et al., 2025), while mechanistic analyses of explicit validation identify attention circuits associated with arithmetic inconsistency (Bertolazzi et al., 2025). More broadly, post-answer representations can support error detection and correction (Kumaran et al., 2026), and uncertainty has been used to guide test-time intervention (Wang et al., 2022). Our setting instead traces natural digit errors as they are read back: matched digit/context counterfactuals localize compatibility-sensitive computation to carry routing, whose proposals combine with late-layer uncertainty to provide a training-free correction signal.

8 Conclusion and Discussion

This paper links the generation and propagation of local arithmetic errors. Weak carry-sensitive influence can produce an off-by-one digit, whose read-back then redirects the next computation through localized lower-/higher-carry routing. Yet committing to the wrong route does not erase arithmetic structure: downstream states still distinguish incompatible from compatible continuations after the erroneous digit has been emitted. Our correction method exploits this one-token-late signal by evaluating candidate branches before commitment.

Our analysis benefits from single-digit tokenization, which provides a direct alignment between generated digits and internal arithmetic states. Multi-digit tokenization may entangle several decimal positions and complicate this correspondence. Moreover, although the broader generation–routing–compatibility pattern recurs across model families, architectures, and addition difficulties (see Appendices D and E), our empirical focus remains addition. Whether analogous generation–routing–compatibility dynamics govern errors in more general reasoning tasks remains an open question.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript organization, language revision, LaTeX preparation, literature search, and code-related development. They were also used to discuss and refine the presentation and interpretation of experimental analyses. All experimental results, methodological decisions, citations, and scientific claims were reviewed and verified by the authors. The authors take full responsibility for the final content of the paper.

ETHICS STATEMENT

This study uses synthetic arithmetic tasks and does not involve human participants or personal data. Its conclusions concern controlled arithmetic behavior and should not be interpreted as evidence of general-purpose error awareness.

REPRODUCIBILITY STATEMENT

We provide detailed model configurations, prompting and dataset construction procedures, counterfactual construction, intervention protocols, evaluation metrics, and cross-model analyses in the appendices. All reported experiments use publicly available model checkpoints with specified revisions and deterministic decoding where applicable. Code for reproducing the experiments and figures will be released publicly upon acceptance.

REFERENCES

Tanja Baeumel, Josef Van Genabith, and Simon Ostermann. The lookahead limitation: Why multi-operand addition is hard for llms. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 250–262, 2025.

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023. URL https://arxiv.org/abs/2303.08112.

Leonardo Bertolazzi, Philipp Mondorf, Barbara Plank, and Raffaella Bernardi. The validation gap: A mechanistic analysis of how language models compute arithmetic but fail to validate it. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 29375–29412, 2025.

Usha Bhalla, Suraj Srinivas, Asma Ghandeharioun, and Himabindu Lakkaraju. Towards unifying interpretability and control: Evaluation via intervention. *arXiv preprint arXiv:2411.04430*, 2024.

Alex Bie, Travis Dick, Alex Kulesza, Prabhakar Raghavan, Vinod Raman, and Sergei Vassilvitskii. Ai-rithmetic. *arXiv preprint arXiv:2602.10416*, 2026.

Hanseul Cho, Jaeyoung Cha, Srinadh Bhojanapalli, and Chulhee Yun. Arithmetic transformers can length-generalize in both operand length and count. In *International Conference on Learning Representations*, volume 2025, pp. 79987–80029, 2025.

Tony Feng, Trieu H Trinh, Garrett Bingham, Dawsen Hwang, Yuri Chervonyi, Junehyuk Jung, Joonkyung Lee, Carlo Pagano, Sang-hyun Kim, Federico Pasqualotto, et al. Towards autonomous mathematics research. *arXiv preprint arXiv:2602.10177*, 2026.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. URL https://arxiv.org/abs/2503.19786.

Dharshan Kumaran, Viorica Patraucean, Simon Osindero, Petar Veličković, and Nathaniel Daw. How llms detect and correct their own errors: The role of internal confidence signals. *arXiv preprint arXiv:2604.22271*, 2026.

Minh-Thang Luong, Dawsen Hwang, Hoang H Nguyen, Golnaz Ghiasi, Yuri Chervonyi, Insuk Seo, Junsu Kim, Garrett Bingham, Jonathan Lee, Swaroop Mishra, et al. Towards robust mathematical reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 35406–35430, 2025.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. URL https://arxiv.org/abs/1802.03426.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9XFSbDPmdW.

Yaniv Nikankin, Anja Reusch, Aaron Mueller, and Yonatan Belinkov. Arithmetic without algorithms: Language models solve math with a bag of heuristics. In *International Conference on Learning Representations*, volume 2025, pp. 55939–55965, 2025.

nostalgebraist. Interpreting GPT: The logit lens. LessWrong, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Philip Quirke and Fazl Barez. Understanding addition in transformers. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=rIx1YXVWZb.

Philip Quirke, Clement Neo, and Fazl Barez. Understanding addition and subtraction in transformers. *arXiv preprint arXiv:2402.02619v9*, 2025. URL https://arxiv.org/abs/2402.02619v9. Version 9, revised February 2025.

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

Qwen Team. Qwen3.8: Model family. Hugging Face model card, 2026b. URL https://huggingface.co/Qwen/Qwen3.8-27B.

Yucheng Sun, Alessandro Stolfo, and Mrinmaya Sachan. Probing for arithmetic errors in language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 8111–8128. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.411. URL https://aclanthology.org/2025.emnlp-main.411/.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Liuyuan Wen, Xun Zhu, Lihao Huang, Wenbin Li, and Yang Gao. The shape of addition: Geometric structures of arithmetic in large language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2606.03645.

Yang Yan, Yu Lu, Renjun Xu, and Zhenzhong Lan. Do large language models truly grasp addition? a rule-focused diagnostic using two-integer arithmetic. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 13478–13494, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chen Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. *arXiv preprint arXiv:2412.06464*, 2024. URL https://arxiv.org/abs/2412.06464.

Tianyi Zhou, Deqing Fu, Vatsal Sharan, and Robin Jia. Pre-trained large language models use fourier features to compute addition. *Advances in Neural Information Processing Systems*, 37: 25120–25151, 2024.

Appendix Contents

A Prompting and Dataset Construction **13**

A.1 Main-Model Setup and Prompting 13

A.2 Natural First-Error Screening 13

A.3 Balanced Carry-Transition Cohorts 14

A.4 Digit-Preserved Carry-Flip Construction 14

A.5 Source-Level Bookkeeping 15

B Generalization across answer positions **16**

C Downstream Compatibility Diagnostics **17**

C.1 Models and Architectural Configurations 17

C.2 Layer-Wise Predictive Entropy 17

C.3 Representation Geometry after Read-Back 17

C.4 Correctness Becomes More Decodable after Read-Back 19

D Mechanisms in Other Models **21**

D.1 Routing Heads in Qwen3-30B and Gemma-3 21

D.2 Gated DeltaNet Models 22

E Details of Inference-Time Correction **24**

E.1 Evaluation Protocol and Metrics 24

E.2 Directional Selection and Entropy Acceptance 24

E.3 Probe Training and Training-required Baselines 26

E.4 Results and Layer Ablation 26

A Prompting and Dataset Construction

A.1 Main-Model Setup and Prompting

Unless otherwise stated, the main-text experiments use Qwen3-4B-Instruct-2507 (Yang et al., 2025). We study addition of three 10-digit integers and use greedy decoding throughout. Answer positions are indexed from left to right, with $p = 0$ denoting the most significant generated digit.

Since the sum of three 10-digit integers can contain either 10 or 11 digits, the same p would otherwise correspond to different decimal significances across examples. We therefore retain only problems whose exact answer has 11 digits, which form the larger subset in our generated pool. Under this restriction, position p always corresponds to decimal significance $k = 10 - p$.

Following Wen et al. (2026), we use the model's standard chat template with an explicit assistant-side arithmetic prefix. The user is instructed to return only the numerical answer, after which the expression, an equals sign, and a space are appended to the assistant prefix before decoding. Thus, the model begins generation directly at the first answer digit. An illustrative example is

Illustrative Prompt and First-Error Generation

```
[User Control Tokens]
Calculate 4837261054 + 6719053288 + 2543187691.
Only output a number.  Don't output commas.

[Assistant Control Tokens]
4837261054 + 6719053288 + 2543187691 = 13999502033
```

In this example, the model outputs **`13999502033`** while the exact sum is `14099502033`. Thus, the model first deviates at $p = 1$, producing `3` instead of the ground-truth digit `4` after correctly generating $d_0 = 1$. We verify that, under this prompt, each digit is represented by a single token, giving a one-to-one correspondence between answer positions and read-back digit tokens.

A.2 Natural First-Error Screening

We first use a separate screening cohort of 10,000 three-operand addition problems to characterize naturally occurring errors (Table 5). Generation is inspected from left to right, and the first position at which the generated digit differs from the exact answer is recorded. Thus, every first-error source has a correct generated prefix: $d_{<p} = d^{\mathrm{GT}}_{<p}, d^{W}_{p} \neq d^{\mathrm{GT}}_{p}$. We call a first error *off-by-one* when $(d^{W}_{p} - d^{\mathrm{GT}}_{p}) \bmod 10 \in \{1, 9\}$.

Table 5: **Natural first-error screening by answer position.**

p	Off-by-one	All first errors	Off-by-one (%)
0	226	235	96.17
1	254	278	91.37
2	189	191	98.95
3	328	330	99.39
4	547	599	91.32
5	504	685	73.58
6	405	447	90.60
7	327	345	94.78
8	209	221	94.57
9	190	214	88.79
10	0	1	0.00
all	3179	3546	89.65

This screening population is used only to estimate the prevalence and position distribution of natural errors; it is distinct from the balanced datasets used for mechanistic interventions. For example, at the main position $p = 1$, 254 of 278 natural first errors are off-by-one.

A.3 Balanced Carry-Transition Cohorts

The mechanistic datasets are stratified by the incoming carry at the target position. We consider the four admissible one-step transitions

$$\mathcal{T} = \{0\to1,\ 1\to0,\ 1\to2,\ 2\to1\},$$

where $a \to b$ means that the ground-truth incoming carry is $c_p = a$ and the off-by-one alternative digit corresponds to the carry hypothesis b. At the boundary position $p = 0$, answer-length constraints make only a subset of these transitions arithmetically feasible; we therefore use only the feasible transition strata there, rather than enforcing the four-way balance below.

For every interior analyzed position $p > 0$, the first-error dataset contains 200 eligible natural first-error sources for each transition:

$$\mathcal{D}_p^{\mathrm{FE}} = \bigcup_{(a\to b)\in\mathcal{T}} \mathcal{D}_{p,a\to b}^{\mathrm{FE}}, \qquad \left|\mathcal{D}_{p,a\to b}^{\mathrm{FE}}\right| = 200. \tag{10}$$

Each retained source satisfies $c_p = a$ and $c_p^W = b$, where c_p^W is the carry hypothesis implied by the naturally generated adjacent digit d_p^W. Each source is subsequently expanded into the matched Cases $\mathcal{A}/\mathcal{B}/\mathcal{C}$ described below.

We construct a separate correct-control dataset with the same transition balance:

$$\mathcal{D}_p^{\mathrm{CC}} = \bigcup_{(a\to b)\in\mathcal{T}} \mathcal{D}_{p,a\to b}^{\mathrm{CC}}, \qquad \left|\mathcal{D}_{p,a\to b}^{\mathrm{CC}}\right| = 200. \tag{11}$$

Each element is a matched pair $(X^{\mathrm{CC}}, X'^{\mathrm{CC}})$ for which both problems are naturally correct through position p, while the incoming carry changes from a to b. The two directions originating from $c_p = 1$ are treated as separate strata, with 200 pairs for $1\to0$ and 200 for $1\to2$.

Table 6: **Balanced datasets at each interior analyzed position.** An FE source yields one matched $\mathcal{A}/\mathcal{B}/\mathcal{C}$ triplet, whereas a CC source yields one original/carry-flipped pair.

Carry transition	FE sources	CC pairs
$0\to1$	200	200
$1\to0$	200	200
$1\to2$	200	200
$2\to1$	200	200
Total	800	800

The balanced cohorts are therefore intervention datasets rather than estimates of the natural carry distribution. In particular, they should not be conflated with the screening frequencies in Table 5.

A.4 Digit-Preserved Carry-Flip Construction

For each first-error source X^{FE}, Case $\mathcal{A}$ is the original natural trajectory, in which the model generates d_p^W after a correct prefix. Case $\mathcal{B}$ keeps the same arithmetic problem and preceding prefix but reads back d_p^{GT} at the target position.

Case $\mathcal{C}$ is obtained from a new arithmetic problem X'^{FE}. We modify only lower-significance operand digits that determine the incoming carry to the target column, while leaving the target raw sum unchanged. A proposed carry flip is retained only if

$$r'_p = r_p, \qquad c'_p = c_p^W, \qquad d_p'^{\mathrm{GT}} = d_p^W, \qquad d_j'^{\mathrm{GT}} = d_j^{\mathrm{GT}} \quad (j \neq p), \tag{12}$$

and the model naturally generates X'^{FE} correctly through position p. The construction therefore changes the objective carry context of the target digit while preserving the answer length, the answer prefix preceding p, and every ground-truth answer digit other than d_p.

The resulting triplet has the form

$$\mathcal{A} = (X^{\mathrm{FE}}, d_p^W), \qquad \mathcal{B} = (X^{\mathrm{FE}}, d_p^{\mathrm{GT}}), \qquad \mathcal{C} = (X'^{\mathrm{FE}}, d_p^W).$$

Consequently, $\mathcal{A}/\mathcal{B}$ changes the read-back digit at fixed arithmetic context, whereas $\mathcal{A}/\mathcal{C}$ preserves the read-back digit while changing its arithmetic compatibility. By construction, all three branches share the same ground-truth continuation at $p+1$.

We also considered a fourth branch,

$$\mathcal{D} = (X'^{\mathrm{FE}}, d_p^{\mathrm{GT}}),$$

which formally completes the crossing of the two arithmetic contexts and two read-back digits. We do not use $\mathcal{D}$ as a symmetric compatibility control, however, because under the carry-flipped context d_p^{GT} is frequently unsupported by either locally available carry route. Such cases therefore change not only arithmetic compatibility but also the routing regime, and behave as off-route interventions rather than as route-supported counterparts of Case $\mathcal{C}$. For this reason, the $\mathcal{A}/\mathcal{B}/\mathcal{C}$ construction is intended as a pair of complementary matched contrasts, not as a 2×2 factorial manipulation of a binary compatibility variable.

Correct-control pairs use the same carry-flip construction. Here, both X^{CC} and its matched carry-flipped counterpart X'^{CC} are naturally generated correctly through p, with $c_p = a$ and $c'_p = b$, respectively. Thus, each pair has the form

$$(X^{\mathrm{CC}}, d_p^{\mathrm{GT}}), \qquad (X'^{\mathrm{CC}}, d_p'^{\mathrm{GT}}).$$

These pairs provide the matched carry-sensitive direction for comparing natural first errors with successful arithmetic computation, without using a naturally erroneous recipient.

A.5 Source-Level Bookkeeping

All derived branches from the same source problem are treated as a single sampling unit. Whenever development and evaluation subsets are formed, an entire $\mathcal{A}/\mathcal{B}/\mathcal{C}$ triplet, or an entire correct-control pair, is assigned to one subset rather than split across them. Component or head selection is performed before evaluating the corresponding localized intervention.

Ground-truth digits, carries, and counterfactual validity checks are used to construct and evaluate the mechanistic datasets. They are not available to the training-free methods at inference time in Section 6.

B GENERALIZATION ACROSS ANSWER POSITIONS

To test whether the routing effect is specific to the main-text position $p = 1$, we repeated the routing-address analysis at representative answer positions $p \in \{0, 3, 6, 9\}$. These positions span the 11-digit answer string and include both boundary cases $p = 0$ and $p = 9$. We do not analyze $p = 10$, because routing is defined through its effect on the next answer position $p + 1$, which does not exist at the final digit.

Figure 7 shows the resulting routing plots. Positions $p = 3, 6, 9$ closely match the main-text pattern. At the boundary position $p = 0$, substantially fewer triplets satisfy the strict routing-resolution criterion, but the retained triplets also show predominantly cross-head $\mathcal{A} \to \mathcal{B}$ and $\mathcal{A} \to \mathcal{C}$ transitions. Across these resolved cases, read-backs that change the digit ($\mathcal{A}/\mathcal{B}$) or change only arithmetic compatibility ($\mathcal{A}/\mathcal{C}$) typically switch between H24 and H27. This replicates the main conclusion that the two heads encode competing lower- and higher-carry routes, and that both digit replacement and compatibility reversal redirect the continuation through the opposite route.

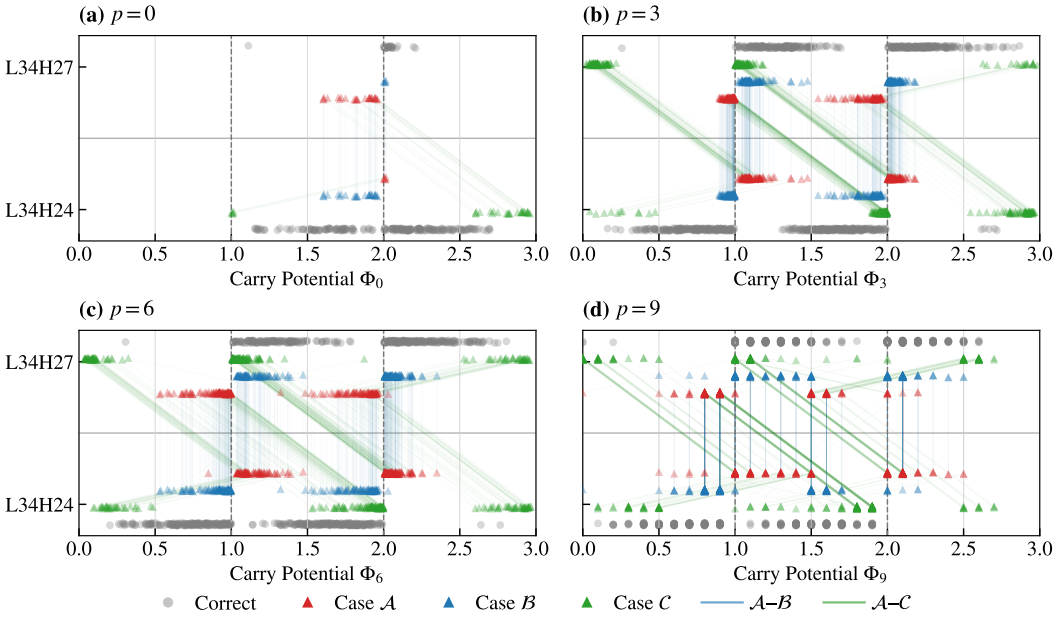

Figure 7: **Routing-address analysis across answer positions** $p \in \{0, 3, 6, 9\}$. Settings are similar to Figure 5.

The main qualification is the lower coverage at the boundary position $p = 0$. Although the resolved triplets remain qualitatively consistent with the routing pattern, many $p = 0$ triplets cannot be uniquely assigned under the strict three-candidate head criterion. One plausible source of this difference is the boundary-specific routing edge: at $p = 0$, the relevant cached key lies at the prompt boundary rather than on a preceding answer digit, so the H24/H27 addressing pattern need not behave identically to interior positions. We therefore treat $p = 0$ as supportive but less conclusive evidence.

A further boundary effect appears for the $\mathcal{C}$ branches of the $1 \to 0$ and $1 \to 2$ transitions (refer to Figure 5). These counterfactuals lie close to the outer limits of the carry-potential range, where only one off-by-one carry direction is arithmetically feasible. We would therefore expect the lower-boundary cases to favor H24 and the upper-boundary cases to favor H27. However, many of these cases instead remain on the opposite head. We interpret this as a natural consequence of the model's continuous internal computation: the routing scores are not explicitly constrained to valid discrete carry values and can occasionally favor an out-of-range carry hypothesis beyond the feasible boundary. These cases therefore do not contradict the lower/higher carry interpretation.

Overall, these results support a position-generalized routing mechanism across representative answer positions.

C Downstream Compatibility Diagnostics

This appendix expands the downstream diagnostics of Section 5. We first describe the model families used in the cross-model analysis, and then examine whether the post-read-back signatures observed in the main model recur across architectures. We consider predictive entropy across layers, representation geometry, and the linear decodability of correctness before and after read-back. From now on, we include all applicable positions unless otherwise specified.

C.1 Models and Architectural Configurations

The main-text analysis uses Qwen3-4B-Instruct-2507. To test whether the downstream signatures depend on this particular architecture, we additionally evaluate Qwen3-30B-A3B-Instruct-2507 (Yang et al., 2025), Qwen3.5-4B (Qwen Team, 2026a), Qwen3.8-27B (Qwen Team, 2026b), and Gemma-3-4B-IT (Gemma Team, 2025). These checkpoints span dense and mixture-of-experts Transformers, hybrid Gated DeltaNet–attention architectures (Yang et al., 2024), and alternating sliding/full-attention architectures. For multimodal checkpoints, our experiments use only the text backbone. Table 7 summarizes the architectural features relevant to the layer-wise analyses below.

Table 7: **Models used for downstream diagnostic analysis.** Q/KV counts refer to full-attention layers. Gated DeltaNet blocks in the Qwen3.5-style hybrid models use a separate recurrent head parameterization. Qwen3-4B-Instruct-2507 is the primary model used throughout the main text.

Model	Layers	Attention pattern	Full-attention layers	Q/KV heads
Qwen3-4B-Instruct-2507	36	Full attention	L1–L36	32/8
Qwen3-30B-A3B-Instruct-2507	48	Full attention & MoE	L1–L48	32/4
Qwen3.5-4B	32	3 Gated DeltaNet + 1 full	L4,8,. . . ,28,32	16/4
Qwen3.8-27B	64	3 Gated DeltaNet + 1 full	L4,8,. . . ,60,64	24/4
Gemma-3-4B-IT	34	5 sliding + 1 full	L6,12,18,24,30	8/4

C.2 Layer-Wise Predictive Entropy

We first examine when the downstream distinction between correct and erroneous continuations becomes visible in the predictive distribution. For each layer L, we apply a standard logit-lens projection (nostalgebraist, 2020; Belrose et al., 2023) to the state at $p+1$, after the digit d_p has been read back. Specifically, we pass the intermediate state through the model's final normalization and unembedding head,

$$z_{p+1}^{(L)} = W_U \,\mathrm{Norm}\Big(h_{p+1}^{(L)}\Big), \qquad \pi_{p+1}^{(L)} = \mathrm{softmax}\Big(z_{p+1}^{(L)}\Big), \tag{13}$$

and compute the full-vocabulary predictive entropy

$$H_{p+1}^{(L)} = -\sum_{v\in\mathcal{V}} \pi_{p+1}^{(L)}(v) \log \pi_{p+1}^{(L)}(v). \tag{14}$$

We then compare the mean entropy of naturally correct trajectories with that of natural first errors at each layer.

Figure 8 reveals a consistent late-layer separation across architectures. For most of the network, the two trajectories have highly similar entropy, indicating that the read-back error is not expressed as a uniformly elevated uncertainty signal throughout computation. The difference instead emerges sharply near the output end: first-error trajectories retain higher entropy than correct trajectories in the late layers of all five models. Thus, the most transferable feature is not a particular layer index, but the emergence of compatibility-sensitive uncertainty late in the computation.

C.3 Representation Geometry after Read-Back

We next ask whether target-position correctness remains visible in the representation after the emitted digit has been read back. Following the visualization protocol of Wen et al. (2026), we project final-layer states with UMAP (McInnes et al., 2018) using cosine distance and include the output-unembedding vectors of digit tokens 0–9 as semantic anchors. The key difference is temporal:

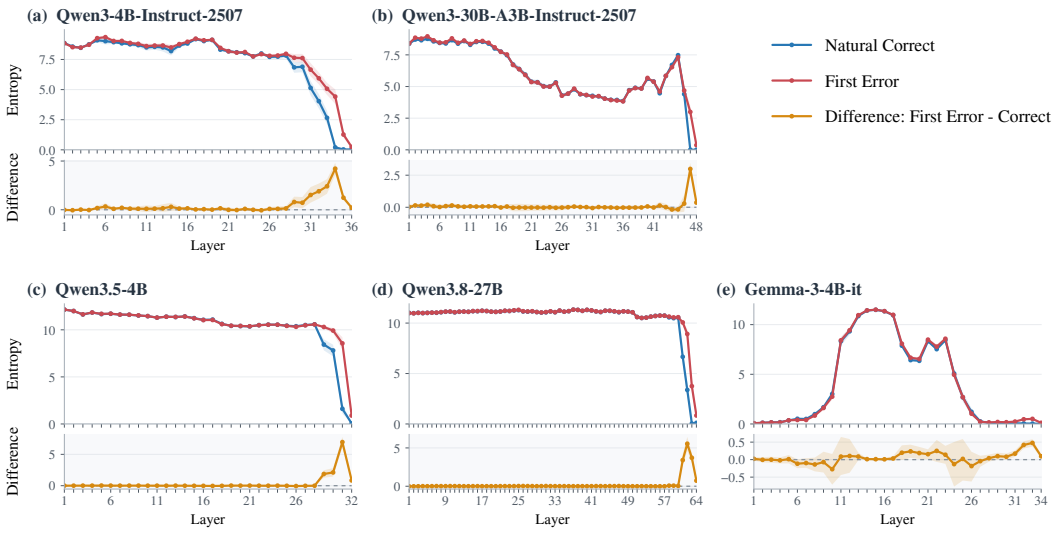

Figure 8: **Layer-wise predictive entropy after read-back across model families.** For each layer, we apply the model's final normalization and unembedding head to the state at $p+1$ and compute full-vocabulary entropy. Blue and red curves show naturally correct and natural first-error trajectories, respectively; the lower panels show their difference, $H_{\mathrm{FE}} - H_{\mathrm{Correct}}$. Across architectures, the strongest separation appears near the output layers.

rather than visualizing $h_p^{(L)}$, we visualize the next-position state $h_{p+1}^{(L)}$, while coloring each point by whether the preceding target digit d_p was naturally correct or a first error.

Figure 9 shows a representative example for Qwen3-4B-Instruct-2507 at $p = 3$. Correct states largely follow the digit-anchored arithmetic manifold, whereas first-error states concentrate in a distinct subregion away from these basins. Thus, after d_p has been read back, the geometry at $p+1$ remains strongly structured by whether the preceding digit was arithmetically compatible. Moreover, many states in this error region also produce an incorrect next digit. A prominent subset consists of $0/9$ confusions, such as $0(9)$ and $9(0)$, consistent with a carry mistake at p propagating into the computation at $p+1$.

Quantifying manifold escape. To quantify the geometric deviation visible in Figure 9, we define a local 5-nearest-neighbor escape ratio in cosine distance. Let

$$\delta_{\cos}(u, v) = 1 - \frac{u^\top v}{\|u\|_2 \|v\|_2}, \tag{15}$$

and let $\mathcal{H}$ denote the set of final-layer next-position states from the clean reference trajectories. For the clean reference h_i^{clean}, define its local manifold scale as

$$d_{\mathrm{clean}}(i) = \frac{1}{5} \sum_{j \in \mathrm{kNN}_5^{\cos}(h_i^{\mathrm{clean}};\, \mathcal{H} \setminus h_i^{\mathrm{clean}})} \delta_{\cos}\left(h_i^{\mathrm{clean}}, h_j^{\mathrm{clean}}\right). \tag{16}$$

For an evaluated state h_i, we analogously measure its distance to the clean manifold,

$$d_{\text{to-clean}}(i) = \frac{1}{5} \sum_{j \in \mathrm{kNN}_5^{\cos}(h_i;\, \mathcal{H} \setminus h_i^{\mathrm{clean}})} \delta_{\cos}\left(h_i, h_j^{\mathrm{clean}}\right), \tag{17}$$

and define

$$R_{\mathrm{escape}}(i) = \frac{d_{\text{to-clean}}(i)}{d_{\mathrm{clean}}(i)}. \tag{18}$$

The paired clean reference is excluded from the neighbor search to avoid a trivial nearest-neighbor match. By construction, values near one correspond to the typical local scale of the clean manifold, whereas values above one indicate geometric escape.

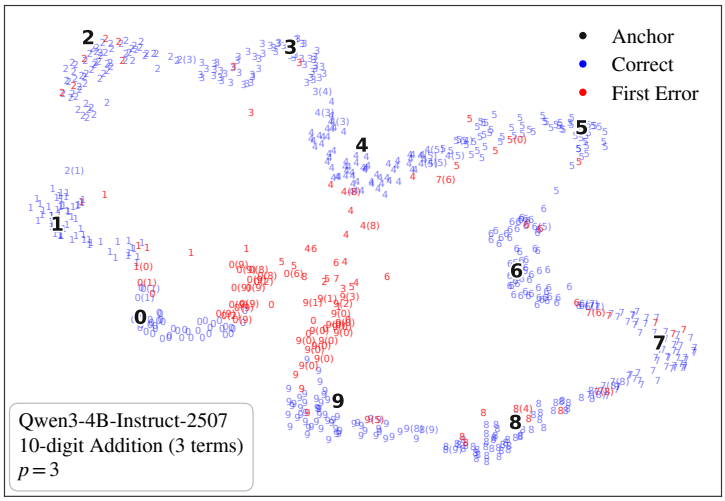

Figure 9: **Next-position representation geometry after read-back.** UMAP of Qwen3-4B-Instruct-2507 final-layer states $h_{p+1}^{(L)}$ at $p = 3$. Black digits are the output-unembedding vectors for tokens 0–9, included as semantic anchors following Wen et al. (2026). Blue and red points indicate whether the preceding target digit d_p was naturally correct or a first error. Each point is labeled by the digit decoded at $p + 1$; when that prediction is incorrect, the ground-truth digit is shown in parentheses, e.g., $0(9)$ means $d_{p+1}^{W} = 0$ while $d_{p+1}^{\mathrm{GT}} = 9$.

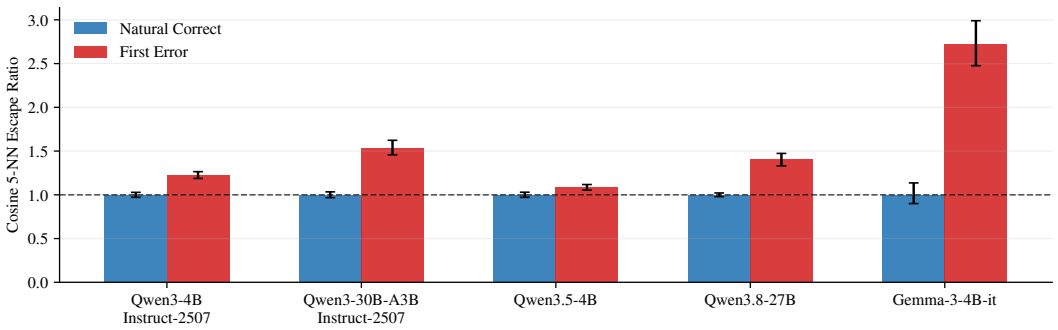

Figure 10: **Post-read-back manifold escape across models.** Mean cosine 5-NN escape ratio of final-layer h_{p+1} states for natural-correct and first-error trajectories. The dashed line at 1 marks the local scale of the clean representation manifold. First-error states lie farther from this manifold across all five evaluated models.

Figure 10 shows the same pattern across all five evaluated models. Natural-correct states remain near the unit reference, while first-error states exhibit consistently larger escape ratios. The magnitude varies across architectures, but the direction is stable. Together with the UMAP visualization, this shows that the post-read-back representation carries a geometric signature of the preceding arithmetic error, complementing the late-layer entropy separation above.

C.4 Correctness Becomes More Decodable after Read-Back

Finally, we ask how explicitly target-position correctness is represented before and after the target digit is emitted. For each model, we train linear probes to predict

$$y_p = \mathbb{1}[d_p = d_p^{\mathrm{GT}}],$$

using either the final-layer of target-position state $h_p^{(L)}$ or the final-layer of next-position state $h_{p+1}^{(L)}$ after d_p has been read back. We report held-out AUROC in Figure 11.

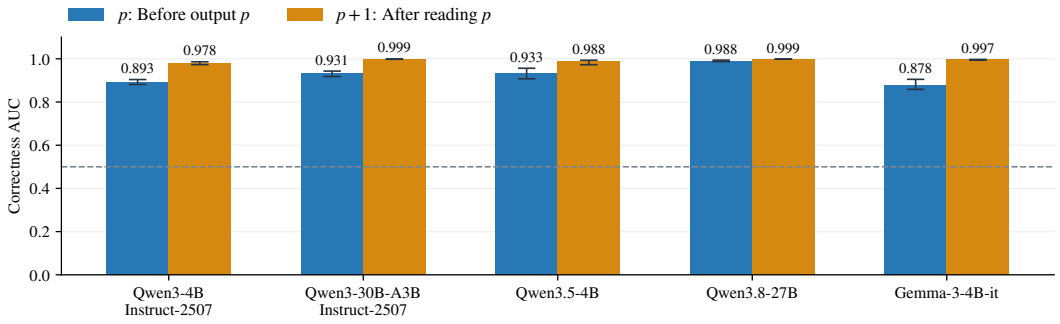

Figure 11: **Target-position correctness is more linearly separable after read-back.** Held-out AUROC of linear probes predicting whether d_p is correct from h_p before emission and from h_{p+1} after d_p has been read back. Correctness is already decodable at the target position, but separability increases consistently at the next position across all five models.

As expected from prior probing studies (Sun et al., 2025; Wen et al., 2026), correctness is already strongly decodable from $h_p^{(L)}$. More strikingly, however, the same label becomes consistently more separable at $p + 1$ across all five model families.

This temporal asymmetry is consistent with the mechanism developed in the main text. Before emission, h_p contains information about the impending target decision; after read-back, the emitted digit has interacted with the cached arithmetic context and leaves a stronger downstream signature of whether that decision was compatible.

This also helps explain why the training-required Next-state probe in Section 6 substantially outperforms the analogous Target-state probe: the post-read-back state provides a more linearly separable correctness signal for revision.

D MECHANISMS IN OTHER MODELS

The main text establishes a lower-/higher-carry routing mechanism in Qwen3-4B-Instruct-2507. We next test how this computation is implemented in four additional models spanning different scales, architectures, and model families. We first consider Qwen3-30B-A3B-Instruct-2507 and Gemma-3-4B-IT, where the routing mechanism remains closely analogous to that of the main model. We then turn to the Qwen3.5-style Gated DeltaNet models, whose implementation differs more substantially.

D.1 ROUTING HEADS IN QWEN3-30B AND GEMMA-3

We identify routing heads using the same $\mathcal{A}/\mathcal{B}$ read-back contrast as in Section 4. For every attention head h at every layer, we perform bidirectional patching and score

$$\Delta\text{Margin}^{\mathcal{A}\leftarrow\mathcal{B}}_{p+1,h} - \Delta\text{Margin}^{\mathcal{B}\leftarrow\mathcal{A}}_{p+1,h}. \tag{19}$$

Large positive values identify heads whose intervention moves both branches toward the donor's continuation. Figure 12 shows the resulting scans. In Qwen3-30B-A3B-Instruct-2507, the two strongest heads are L47H3 and L47H5; in Gemma-3-4B-IT, they are L31H4 and L31H5.

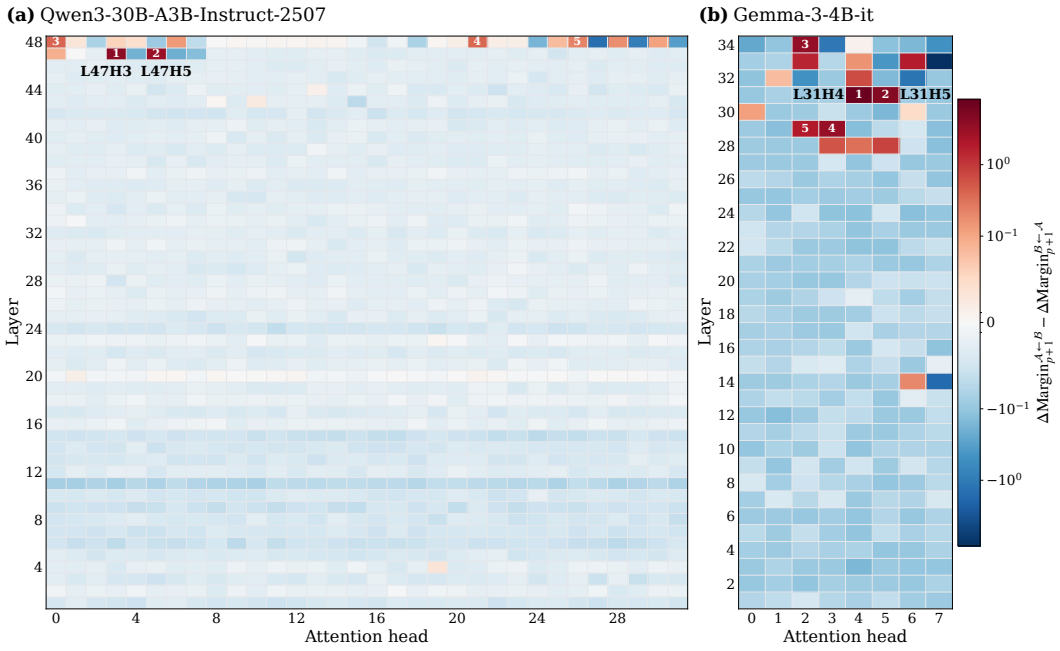

Figure 12: **Cross-model localization of carry-routing heads.** Bidirectional $\mathcal{A}/\mathcal{B}$ head-wise patching scores $\Delta\text{Margin}^{\mathcal{A}\leftarrow\mathcal{B}}_{p+1,h} - \Delta\text{Margin}^{\mathcal{B}\leftarrow\mathcal{A}}_{p+1,h}$ for (**a**) Qwen3-30B-A3B-Instruct-2507 and (**b**) Gemma-3-4B-IT. The two highest-scoring heads are annotated in each model.

We next determine the arithmetic role of each selected head. Similar to Section 4, for every Case $\mathcal{A}$ example, we separately read back the three local candidates $d \in \{d_p - 1, d_p, d_p + 1\}$ and select the digit maximizing that head's target-edge score. We then measure how often the naturally wrong digit is selected separately on under-carry and over-carry errors. A head that preferentially selects the wrong digit on under-carry errors corresponds to the lower-carry route, whereas the converse identifies the higher-carry route.

For Qwen3-30B-A3B-Instruct-2507, L47H3 selects the wrong digit in 100% of under-carry cases and 0% of over-carry cases, identifying it as the *lower-carry* head. L47H5 shows the complementary preference, selecting the wrong digit in 86.67% of over-carry cases but only 4% of under-carry cases, and is therefore the *higher-carry* head.

Gemma-3-4B-IT exhibits the same organization with reversed head identities. L31H5 selects the wrong digit in 92.95% of under-carry cases and 1.11% of over-carry cases, identifying the *lower-*

carry route. L31H4 instead selects it in 97.26% of over-carry cases and 4.14% of under-carry cases, identifying the *higher-carry route*.

Thus, despite differences in model family and architecture, both models recover the same functional decomposition observed in Qwen3-4B: two late attention heads preferentially route the locally lower and higher carry alternatives. The layer and head identities differ, but the carry-relative roles are preserved.

D.2 Gated DeltaNet Models

Qwen3.5-4B and Qwen3.8-27B use Gated DeltaNet blocks, so the attention-edge mechanism identified in the Transformer models above need not transfer literally. We use Qwen3.5-4B as a representative model for fine-grained causal analysis and ask where, after read-back, the observed digit interacts with the preceding arithmetic context to affect the next prediction. We use the same Cases $\mathcal{A}/\mathcal{B}/\mathcal{C}$ defined in Section 2.2: $\mathcal{A}/\mathcal{B}$ changes the read-back digit at fixed context, whereas $\mathcal{A}/\mathcal{C}$ preserves the digit while changing its arithmetic compatibility.

In Qwen3.5-4B, each DeltaNet state head updates a 128×128 recurrent matrix as

$$\begin{aligned} \bar{S}_t &= e^{g_t} S_{t-1}, \\ \delta_t &= \beta_t \left(v_t - \bar{S}_t^\top k_t \right), \\ W_t &= k_t \otimes \delta_t, \\ S_t &= \bar{S}_t + W_t, \\ o_t &= S_t^\top q_t / \sqrt{128}. \end{aligned} \tag{20}$$

Here $\bar{S}_t$ is the decayed pre-token memory and W_t is the current-token rank-1 write. For post-state interventions, we patch the recurrent state inside the DeltaNet kernel, recompute the recipient readout $S_t^\top q_t$, and continue the remaining forward pass.

Layer localization. Figure 13 shows a bidirectional $\mathcal{A} \leftrightarrow \mathcal{B}$ component scan at $p = 3$. As $\mathcal{A}$ and $\mathcal{B}$ share the same pre-read-back context, their pre-token recurrent and convolution states are identical, isolating the effect of the observed digit. As in the Transformer models, we observe an early read-in effect followed by a distinct late stage. The latter concentrates in L29–L31 and is much weaker under FFN patching, motivating a direct analysis of the DeltaNet recurrent update.

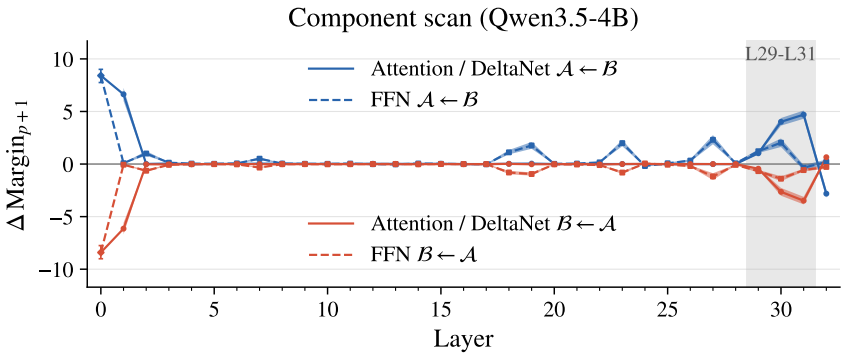

Figure 13: **Component scan in Qwen3.5-4B at $p = 3$.** Bidirectional $\mathcal{A} \leftrightarrow \mathcal{B}$ patching localizes a late causal effect in the L29–L31 DeltaNet blocks.

Digit and context meet in the current-token write. Table 8 zooms into L29–L31. For $\mathcal{A} \leftrightarrow \mathcal{B}$, patching the post-update state S_t changes the next-digit margin by $+0.907/-0.741$, and patching only the rank-1 write W_t reproduces essentially the same effect $(+0.907/-0.791)$. Thus, under fixed arithmetic context, the causal effect of changing the read-back digit is carried almost entirely by the current-token write.

For $\mathcal{A} \leftrightarrow \mathcal{C}$, the observed digit is held fixed and only arithmetic compatibility changes. Again, patching the post-update state S_t has a large bidirectional effect $(+8.636/-8.043)$, and patching

Table 8: **Component-level patching in the late DeltaNet region.** Entries are mean changes in the next-position ground-truth margin. $\mathcal{A} \leftrightarrow \mathcal{B}$ changes the read-back digit under fixed arithmetic context, whereas $\mathcal{A} \leftrightarrow \mathcal{C}$ keeps the observed digit fixed while changing the arithmetic context in which it is interpreted. In both contrasts, the dominant late causal object is the current-token write W_t; short convolution provides its main upstream arithmetic-context input.

Patched quantity	**Layer(s)**	$\mathcal{A} \leftarrow \mathcal{C}$	$\mathcal{C} \leftarrow \mathcal{A}$	$\mathcal{A} \leftarrow \mathcal{B}$	$\mathcal{B} \leftarrow \mathcal{A}$
post-update S_t	L29	+1.145	-0.382	+0.494	-0.115
post-update S_t	L30	+3.747	-1.890	+0.632	-0.193
post-update S_t	L31	+4.350	-3.146	-0.336	-0.150
post-update S_t	**L29–L31**	**+8.636**	-8.043	**+0.907**	-0.741
pre-token S_{t-1}	L29–L31	-0.002	-0.003	0.000	0.000
decay term $\bar{S}_t$	L29–L31	-0.034	-0.017	-0.001	+0.024
rank-1 write W_t	L29–L31	**+8.583**	-8.076	**+0.907**	-0.791
δ_t-only write (recipient k_t, donor δ_t)	L29–L31	**+7.474**	-6.360	**+0.890**	-0.769
pre-token conv state	L29–L31	**+8.619**	-8.021	0.000	0.000

only the write W_t nearly reproduces it ($+8.583/-8.076$). Hence the same current-token write also mediates the compatibility effect. Patching δ_t alone already captures most of this late effect, indicating that the update content is the main carrier. Finally, patching the pre-token convolution state has essentially no effect for $\mathcal{A}/\mathcal{B}$ but nearly reproduces the full $\mathcal{A}/\mathcal{C}$ effect, identifying short convolution as the main upstream context path into the write.

We also find that Qwen3.8-27B exhibits the same stage-level organization, with the corresponding late recurrent effect concentrated in L61–L63.

Inference-time correction. The recurrent-write analysis above provides an architecture-specific proposal signal for Gated DeltaNet models. Let d_p be the model's greedy digit. Starting from the same prefix state, we tentatively read back the local candidates $\{d_p - 1, d_p, d_p + 1\}$. For each candidate d, we compute the ungated DeltaNet innovation

$$\epsilon_{l,h}(d) = v_{l,h}(d) - \bar{S}_{l,h}(d)^\top k_{l,h}(d), \tag{21}$$

and aggregate its energy over the compatibility-sensitive L29–L31 region:

$$E(d) = \sum_{l=29}^{31} \sum_{h} \|\epsilon_{l,h}(d)\|_2^2 . \tag{22}$$

We use this signal only to propose a local carry alternative,

$$d_{\text{route}} = \arg \min_{d \in \{d_p - 1, d_p + 1\}} E(d), \qquad \mathcal{C}_p(d_p) = \{d_p, d_{\text{route}}\}. \tag{23}$$

Thus, the DeltaNet proposal selects a direction among the two adjacent carry alternatives rather than deciding whether the original greedy digit is correct.

For each $d \in \mathcal{C}_p(d_p)$, we compute the next-position full-vocabulary entropy $H_{p+1}^{(L)}(d)$ at $L \in \mathcal{L}$. Here, $\mathcal{L} = \{\text{L30}, \text{L31}, \text{L32}\}$ for Qwen3.5-4B and $\mathcal{L} = \{\text{L62}, \text{L63}, \text{L64}\}$ for Qwen3.8-27B[6]. We replace the greedy digit only when the proposed route has lower entropy at all three layers:

$$\widetilde{d}_p = \begin{cases} d_{\text{route}}, & H_{p+1}^{(L)}(d_{\text{route}}) < H_{p+1}^{(L)}(d_p), \quad \forall L \in \mathcal{L}, \\ d_p, & \text{otherwise.} \end{cases} \tag{24}$$

Hence, the recurrent update proposes the locally relevant competing continuation, while downstream uncertainty determines whether to accept it.

[6]The entropy readout window is intentionally shifted one layer downstream from the recurrent region used for mechanism localization (L29–L31 and L61–L63, respectively). The recurrent region identifies where the digit–context interaction is written, whereas correction reads out its downstream consequence on the predictive distribution. We keep the same three-layer width while including the final downstream layer.

E Details of Inference-Time Correction

The correction experiments serve as a functional validation of the identified mechanisms rather than as a deployment-oriented arithmetic system. Their purpose is to test whether routing and post-read-back compatibility signals are behaviorally actionable, so the baseline comparisons should be interpreted as accuracy references rather than equal-compute benchmarks.

E.1 Evaluation Protocol and Metrics

For each model, we select the operand width on a separate calibration set to obtain greedy-decoding question accuracy near 60%. The width is fixed before correction evaluation, and all methods are evaluated on the same 1,000 held-out questions for that model. Calibration targets approximate baseline difficulty rather than identical test accuracy. Methods generate their own prefixes. Training-free correction acts at $p = 1, \ldots, P-2$; the training-required suite acts at all answer positions. All P answer positions are scored, including boundary digits. Decoding stops after P positions or at EOS; non-digit outputs and positions missing after early EOS are counted as incorrect.

Let y_{ip} be the target digit, o_{ip} Original's output, g_{ip}^m method m's current-prefix greedy proposal, and $\widetilde{d}_{ip}^m$ its committed token. For $T = \sum_i P_i$,

$$\begin{aligned}
\mathrm{TokenAcc}(m) &= T^{-1} \sum_{i,p} \mathbf{1}[\widetilde{d}_{ip}^m = y_{ip}], \\
\mathrm{QuestionAcc}(m) &= N^{-1} \sum_{i} \mathbf{1}[\widetilde{\boldsymbol{d}}_i^m = \boldsymbol{y}_i], \\
\mathrm{ModifiedRate}(m) &= T^{-1} \sum_{i,p} \mathbf{1}[\widetilde{d}_{ip}^m \neq g_{ip}^m].
\end{aligned} \tag{25}$$

Repair and preservation are defined relative to Original:

$$\mathrm{TPcorr}(m) = \frac{\sum_{i,p} \mathbf{1}[o_{ip} \neq y_{ip},\ \widetilde{d}_{ip}^m = y_{ip}]}{\sum_{i,p} \mathbf{1}[o_{ip} \neq y_{ip}]}, \quad \mathrm{FPpres}(m) = \frac{\sum_{i,p} \mathbf{1}[o_{ip} = y_{ip},\ \widetilde{d}_{ip}^m = y_{ip}]}{\sum_{i,p} \mathbf{1}[o_{ip} = y_{ip}]}. \tag{26}$$

These rates include downstream effects of changed prefixes; modified rate counts only local changes to the current proposal.

E.2 Directional Selection and Entropy Acceptance

Table 9 lists the fixed direction signals and entropy layers. Candidate selection uses no ground-truth digits or carries. Write $[d]_{10} = d \bmod 10$ and define

$$\mathcal{N}_p = \{[g_p - 1]_{10}, g_p, [g_p + 1]_{10}\}. \tag{27}$$

Table 9: **Direction signals and entropy readout layers**. Head assignments use a three-candidate fallback when unresolved; recurrent selection always supplies one neighbor.

Model	Direction signal	Entropy layers
Qwen3-4B-Instruct-2507	Low L34H24; high L34H27	L34–L36
Qwen3-30B-A3B-Instruct-2507	Low L47H3; high L47H5	L47–L48
Qwen3.5-4B	Recurrent innovation, L29–L31	L30–L32
Qwen3.8-27B	Recurrent innovation, L61–L63	L62–L64
Gemma-3-4B-IT	Low L31H5; high L31H4	L29–L32

Head-based selection. Each head selects its highest-scoring digit in $\mathcal{N}_p$, breaking ties by digit value. If exactly one head selects g_p, its identity sets R_p; otherwise $R_p = \perp$. A low assignment suggests under-carry and a $+1$ alternative; a high assignment suggests over-carry and a -1 alternative:

$$\mathcal{C}_p = \begin{cases} \{g_p, [g_p + 1]_{10}\}, & R_p = \text{low}, \\ \{g_p, [g_p - 1]_{10}\}, & R_p = \text{high}, \\ \mathcal{N}_p, & R_p = \perp. \end{cases} \tag{28}$$

The direction does not depend on the other head's preferred digit. Scores are compared within each head, including when the two heads lie at different layers.

Recurrent selection. For Qwen3.5 and Qwen3.8, read each candidate from the same prefix and aggregate recurrent innovation (Eq. (21)) over the layers $\mathcal{R}_m$ in Table 9:

$$E_p(d) = \sum_{\ell \in \mathcal{R}_m, h} \|\epsilon_{\ell h, p}(d)\|_2^2. \tag{29}$$

Qwen3.5 reconstructs innovation as $\delta / \max(\beta, 10^{-8})$; Qwen3.8 captures $v - \bar{S}^\top k$ directly. Choose the lower-energy neighbor, with ties favoring -1:

$$\sigma_p = \arg\min_{\sigma \in \{-1,+1\}} \big(E_p([g_p + \sigma]_{10}), \sigma\big), \qquad \mathcal{C}_p = \{g_p, [g_p + \sigma_p]_{10}\}. \tag{30}$$

Entropy acceptance. Apply the model's final normalization and output transformation to each intermediate read-back state; use final logits directly at the last layer. For the resulting full-vocabulary distribution,

$$H_{p+1}^{(\ell)}(d) = -\sum_{v \in \mathcal{V}} \pi_{p+1}^{(\ell)}(v \mid d) \log \pi_{p+1}^{(\ell)}(v \mid d). \tag{31}$$

Head routing accepts a common entropy minimizer over $\mathcal{C}_p$ only if all readout layers agree, breaking ties by digit value. Recurrent correction accepts its neighbor only if its entropy is strictly lower at every layer. Otherwise, retain g_p.

Entropy-only. This baseline uses all three candidates in $\mathcal{N}_p$ without direction signals. All layers must agree on the minimum-entropy candidate; exact ties favor g_p, then $[g_p - 1]_{10}$, then $[g_p + 1]_{10}$.

Algorithm 1 summarizes cached route decoding. Every eligible position uses three independent read-backs. The current training-required and Entropy-only suite instead recomputes the full prefix with caching disabled and appends only the selected token.

Algorithm 1 Cached route selection and entropy correction

Input: model M_θ, prompt X, length P, policy τ, configured layers and heads.
$(\kappa, z) \leftarrow \mathrm{Prefill}_\theta(X)$; $\widetilde{\boldsymbol{d}} \leftarrow ()$
for $p = 0, \ldots, P-1$ **do**
  $g \leftarrow \arg\max z$; $\widetilde{d} \leftarrow g$
  if $1 \le p \le P-2$ and g and its predecessor are digits **then**
    $\mathcal{N} \leftarrow \{[g-1]_{10}, g, [g+1]_{10}\}$
    for each $d \in \mathcal{N}$ **do**
      $(\kappa^{(d)}, z^{(d)}, s^{(d)}) \leftarrow \mathrm{Read}_\theta(\mathrm{Clone}(\kappa), d)$
      $H_\ell(d) \leftarrow \mathrm{Entropy}_\ell(s^{(d)}), \ell \in \mathcal{L}$
    end for
    if $\tau = \mathrm{head}$ **then**
      $\mathcal{C} \leftarrow \mathrm{HeadCandidates}(g, s)$ (Eq. (28))
      $w_\ell \leftarrow \arg\min_{d \in \mathcal{C}}(H_\ell(d), d), \ell \in \mathcal{L}$
      if all w_ℓ agree **then** $\widetilde{d} \leftarrow w_\ell$
    else
      $d' \leftarrow [g + \sigma_p]_{10}$ (Eq. (30))
      if $\forall \ell : \ H_\ell(d') < H_\ell(g)$ **then** $\widetilde{d} \leftarrow d'$
    end if
    $(\kappa, z) \leftarrow (\kappa^{(\widetilde{d})}, z^{(\widetilde{d})})$
  else $(\kappa, z) \leftarrow \mathrm{Read}_\theta(\kappa, g)$
  end if
  $\widetilde{\boldsymbol{d}} \leftarrow \widetilde{\boldsymbol{d}} \,\|\, (\widetilde{d})$
end for; **return** $\widetilde{\boldsymbol{d}}$

E.3 Probe Training and Training-required Baselines

The cross-model suite uses 4,000 training and 1,000 validation questions from the original generation protocol, separate from the frozen test set and mechanism cohorts. Target-state, raw-sum, carry, and linear probes retain correct positions and the first error per natural trajectory; next-state probes use all positions, including a final-digit read-back. MLPs use hidden widths 512/128, ReLU, dropout 0.2, and Adam (10^{-3}; batch size 512), with at most 100 epochs and validation patience eight. Classification uses cross-entropy and carry regression uses squared error. The main-model panel retains its reference-run results.

Original. Original uses full-vocabulary greedy decoding.

Re-prompting. For re-prompting, a target-state MLP disagreement triggers one retry after appending the proposal and "That step looks incorrect. Let's re-do just this step: {expression} = {prefix}". Only the retry's greedy token is committed; the prompt contains no target answer.

Steering. A ten-class linear probe selects a direction added to the final decoder block's raw residual h_p:

$$j_p = \arg\max_j (W h_p + b)_j, \quad h'_p = h_p + \alpha W_{j_p}, \quad \widetilde{t}_p = \arg\max_{t \in \mathcal{V}} \mathrm{Readout}_\theta(h'_p)_t. \tag{32}$$

Only the last input token is edited; readout uses the model's native final normalization and logit transformations. Select $\alpha \in \{0, 0.25, 0.5, 1, 2, 4, 8, 16\}$ on validation natural states by question accuracy, then token accuracy, then smaller gain.

Target-state probe The target-state MLP replaces the proposal with its argmax from the final normalized pre-digit state.

Dual-stream probe. The dual-stream method of Wen et al. (2026) combines raw-column residue and carry-potential probes, with validation-selected layers and consistency tolerance $\delta = 0.1$.

Next-state probe. An MLP is trained to predict the ground-truth target digit from the final normalized state after reading back the model's proposed digit. At inference, we read back the greedy proposal d_p once and directly select $\tilde{d}_p = \arg\max_{k \in \{0,\ldots,9\}} q_p(k \mid d_p)$. We commit $\tilde{d}_p$ and continue decoding from the corrected prefix, updating the cache accordingly if the digit changes.

E.4 Results and Layer Ablation

Table 11 reports the five-model comparison. Route+Entropy uses head identity; Recurrent+Entropy uses recurrent innovation. On the main model, routing improves over Entropy by 3.8 question-accuracy points. Table 10 reports the separate readout-layer ablation: L36 outperforms L33 and L34 alone, while adding L33 to L34/L35/L36 reduces question accuracy from 81.7% to 78.3%.

Table 10: **Main-model Route+Entropy readout-layer ablation on 1,000 questions**. All metrics are percentages.

Readout layers	Token Acc.	Question Acc.	TP corr.	FP pres.
Original	92.07	60.9	–	–
L33	71.80	4.8	52.75	73.44
L34	89.66	40.9	69.04	91.44
L35	94.45	71.2	69.15	96.62
L34/L35	95.44	75.1	70.07	97.62
L36	96.38	80.4	67.55	98.86
L34/L36	96.40	81.5	67.43	98.89
L35/L36	96.56	81.4	67.32	99.08
L34/L35/L36	**96.57**	**81.7**	63.30	99.44
L33/L34/L35/L36	96.09	78.3	57.11	99.45

Table 11: Performance across correction methods. Each task adds three equal-width integers; W and P denote operand width and answer length. All metrics are percentages; Mod. is modified rate. Training-free methods act at $1, \ldots, P-2$; the current training-required suite acts at all answer positions. All P positions are scored. Dashes denote unreported or inapplicable values. Boldface follows the reported within-group accuracy comparisons.

Method	Token Acc.	Question Acc.	Mod.	TP corr.	FP pres.
Qwen3-4B-Instruct-2507; W=10, P=11; three operands					
Original	92.07	60.9	–	–	–
Re-prompting	92.17	61.4	0.13	1.49	99.98
Steering	95.50	63.5	0.35	45.53	99.80
Target-state probe	96.09	66.4	4.49	54.13	99.70
Dual-stream probe	93.71	59.3	3.55	32.68	98.96
Next-state probe	**97.96**	**84.4**	4.15	80.28	99.49
Entropy	95.82	77.9	4.29	50.46	99.72
Route+Entropy	**96.57**	**81.7**	4.57	63.30	99.44
Qwen3-30B-A3B-Instruct-2507; W=20, P=21; three operands					
Original	94.50	54.8	–	–	–
Re-prompting	94.63	56.3	0.10	3.90	99.91
Steering	94.66	54.4	0.22	5.19	99.87
Target-state probe	96.27	55.8	2.72	49.18	99.01
Dual-stream probe	96.20	54.7	2.11	39.65	99.49
Next-state probe	**99.09**	**86.2**	2.95	89.44	99.65
Entropy-only	**99.24**	87.6	2.84	90.30	99.76
Route+Entropy	99.17	**88.0**	2.82	89.96	99.71
Qwen3.5-4B; W=15, P=16; three operands					
Original	93.17	63.5	–	–	–
Re-prompting	93.64	65.3	0.12	7.78	99.93
Steering	95.21	63.8	2.74	46.66	98.77
Target-state probe	95.69	65.6	2.98	52.70	98.84
Dual-stream probe	95.21	62.6	2.76	44.10	98.96
Next-state probe	**97.54**	**81.3**	2.79	72.92	99.34
Entropy-only	96.50	77.8	1.70	52.88	99.70
Recurrent+Entropy	**96.61**	**77.9**	1.84	55.17	99.65
Qwen3.8-27B; W=24, P=25; three operands					
Original	92.30	68.8	–	–	–
Re-prompting	92.36	68.4	0.02	0.99	99.98
Steering	92.28	68.5	0.06	0.21	99.96
Target-state probe	95.69	69.2	2.01	53.30	99.23
Dual-stream probe	91.19	29.3	6.98	56.00	94.13
Next-state probe	**96.40**	**82.0**	2.16	61.66	99.30
Entropy-only	94.76	81.9	1.35	37.45	99.54
Recurrent+Entropy	**94.77**	**81.9**	1.32	35.43	99.72
Gemma-3-4B-IT; W=8, P=9; three operands					
Original	81.20	55.9	–	–	–
Re-prompting	81.72	56.8	0.26	2.90	99.97
Steering	81.20	55.9	0.00	0.00	100.00
Target-state probe	90.68	61.8	6.20	58.27	98.18
Dual-stream probe	82.27	39.4	17.41	67.79	85.62
Next-state probe	**95.76**	**78.2**	5.88	81.32	99.10
Entropy-only	92.47	70.3	5.60	69.44	97.80
Route+Entropy	**94.71**	**78.2**	5.93	78.19	98.54