ROUTEVERIFY: ROUTING INTERVENTIONS FOR STEP-LEVEL VERIFICATION IN MIXTURE-OF-EXPERTS MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Routing states in mixture-of-experts (MoE) models can distinguish correct from erroneous reasoning steps, but such information does not by itself show whether changing routing changes a model's step-level verification judgment. We introduce **RouteVerify**, a controlled intervention study that replaces selected internal activations from one solution trace with activations from another, while keeping the evaluated input unchanged. Across nine MoE checkpoints, we compare three intervention interfaces: router logits, MoE outputs, and post-MoE residual states, and vary the layers and token positions at which activations are replaced. We measure changes in a verification margin that quantifies the model's preference for `INCORRECT` over `CORRECT`. We also evaluate the same input with activations from correct versus erroneous steps to test whether the resulting judgment depends on the source step's correctness. At the same layer and token positions, post-MoE residual replacements produce larger paired margin effects than router-logit or MoE-output replacements. Router-logit interventions become substantially stronger when extended across multiple routed layers and tokens, and intervening on equal numbers of tokens at different positions produces systematically different effects. However, large paired effects do not necessarily imply consistent differences between correct and erroneous source steps. With the evaluated input fixed, erroneous-step activations produce more `INCORRECT`-favoring margins than correct-step activations under local residual interventions in most checkpoints; this ordering is less consistent under broad routing interventions. These findings distinguish changing a verification judgment from changing it according to the source step's correctness, and show that both depend on the intervention interface and its layer and token coverage.

1 INTRODUCTION

Process verification asks a model to determine whether an intermediate reasoning step is correct, rather than only judging the final answer (Lightman et al., 2024). Recent work shows that internal representations can reveal output correctness (Orgad et al., 2025), while hidden-state interventions can also change step-level verification behavior (Zhou et al., 2026). In mixture-of-experts (MoE) models, routing provides an additional intervention interface through expert selection and weighting. This raises a basic question: when and where do routing interventions matter for step-level verification?

We investigate this question with ROUTEVERIFY (Figure 1), a controlled activation-replacement study across nine MoE checkpoints. We use paired solution traces for the same problem: one is correct, and the other contains its first error at the step being evaluated. The model is asked to judge that step as `CORRECT` or `INCORRECT`. We record internal activations while the model processes one trace and replace selected activations in its computation on the other trace. The example supplying the activations is the *donor*; the example whose activations are replaced is the *receiver*. The receiver's input text remains unchanged, and we measure the effect on its verification judgment.

We quantify judgment changes using a *verification margin*: the difference between the model's length-normalized log probabilities for `INCORRECT` and `CORRECT`, with larger values favoring

Code: `https://anonymous.4open.science/r/RouteVerify-A370/`.

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.

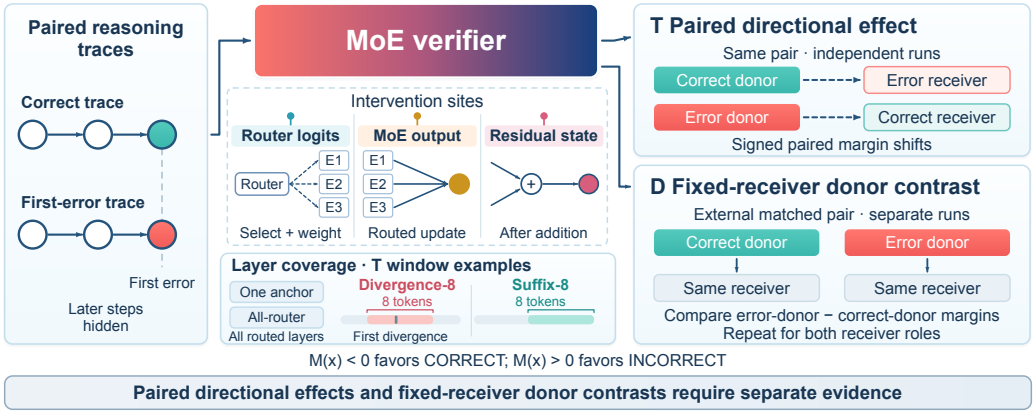

Figure 1: Overview of RouteVerify. Activations from a donor example replace selected activations in a receiver example, whose input text remains unchanged and whose verification judgment is measured. We vary the intervention interface, layers, and token positions. T summarizes the signed margin changes from the two opposite-direction replacements within a correct/error pair. D compares activations from correct and erroneous donors from another problem while holding the receiver fixed. Positive D means that the erroneous donor produces a more INCORRECT-favoring margin than the correct donor.

INCORRECT. A large change in this margin, however, does not necessarily mean that the model responds to whether the donor step is correct. For example, suppose the model initially favors the appropriate label for each member of a correct/error pair. An intervention that pushes both judgments toward neutrality moves each margin toward the other example's label, even if the effect does not depend on the donor step's correctness. We therefore distinguish changes produced by exchanging activations within a pair from differences produced by correct versus erroneous donors on the same receiver. For the latter comparison, we keep the receiver fixed and obtain both donors from a matched pair associated with another problem.

We compare three intervention interfaces: *router logits*, which score candidate experts; *MoE outputs*, which contain the routed experts' combined contribution; and *post-MoE residual states*, which contain the representation after the MoE residual update. We vary both the layers and the token positions at which activations are replaced. A *token window* specifies the selected positions within the evaluated reasoning step, such as its final token, its final eight tokens, or the entire step. This design allows us to examine both the location and the extent of an intervention.

Our results show strong dependence on these choices. At the same layer and token window, post-MoE residual replacements produce larger paired margin effects than router-logit or MoE-output replacements in all 36 comparisons across nine checkpoints and four windows. Router-logit interventions become substantially stronger when expanded across routed layers and tokens, while comparing two eight-token windows at different positions reveals systematic position dependence. Yet these paired effects do not necessarily predict how a fixed receiver responds to donor correctness. Local residual interventions produce more INCORRECT-favoring margins with erroneous than with correct donors in most checkpoints, whereas broad routing interventions show less consistent differences across checkpoints and correct versus erroneous receivers. Together, these findings show that changing a verification judgment and changing it according to the donor step's correctness are distinct outcomes, both of which depend on where and how broadly the intervention is applied.

2 Intervention Design

2.1 Verification task and score

Our evaluation set is derived from PRMBench, a benchmark with step-level annotations for evaluating fine-grained error detection in reasoning traces (Song et al., 2025). We construct 290 pairs of solution traces. In each pair, one trace is labeled correct and the other contains its first annotated error at the

evaluated step. The model receives the problem, the solution trace through this step, and a fixed instruction to judge the step as `CORRECT` or `INCORRECT`. Later steps and analysis metadata are not provided. We write C and E for the correct and error-bearing members of a pair, respectively.

Rather than recording only which label the model prefers, we measure its relative preference using a continuous *verification margin*:

$$M(x) = |y_{\mathrm{I}}|^{-1} \log p(y_{\mathrm{I}} \mid x) - |y_{\mathrm{C}}|^{-1} \log p(y_{\mathrm{C}} \mid x), \tag{1}$$

where y_{I} and y_{C} are the model-specific token sequences for `INCORRECT` and `CORRECT`, and $|y|$ denotes the number of tokens in a sequence. Positive values favor `INCORRECT`; negative values favor `CORRECT`. An increase in the margin therefore indicates a shift toward an `INCORRECT` judgment, although it need not change the preferred label. We use average log probability per token so that the comparison does not directly reflect the summation of different numbers of token log probabilities. We also report total-logprob margins as a sensitivity analysis.

2.2 Activation replacement and two intervention measures

Donors, receivers, and replacement. We call the example supplying recorded activations the *donor* and the example whose activations are replaced the *receiver*. The receiver retains its original problem, solution text, and verification instruction. Only the selected internal activations are replaced; the model then continues its computation and scores the two judgment labels.

For an intervention setting s, let h_s^r and h_s^d denote the corresponding receiver and donor activations. We replace the receiver activations according to

$$\widetilde{h}_s^r = h_s^r + \alpha(h_s^d - h_s^r), \tag{2}$$

where α controls the replacement strength. All new experiments use $\alpha = 1$, corresponding to exact replacement. The setting s specifies the intervention interface and its layer and token coverage. Let $x_{r \leftarrow d,s}$ denote the receiver evaluated under this intervention. Its margin change is

$$\Delta_{d \to r,s} = M(x_{r \leftarrow d,s}) - M(x_r). \tag{3}$$

Paired directional effect, T. For each correct/error pair, we perform two replacements: activations from the erroneous trace are inserted into the correct receiver, and activations from the correct trace are inserted into the erroneous receiver. We summarize these two changes as

$$T_s = \tfrac{1}{2} \left(\Delta_{\mathrm{E} \to \mathrm{C},s} - \Delta_{\mathrm{C} \to \mathrm{E},s} \right). \tag{4}$$

The signs are chosen so that an increase in the correct receiver's margin and a decrease in the erroneous receiver's margin both contribute positively. Thus, positive T indicates an average shift toward the donor step's label across the two replacements; it does not require both replacements to move in that direction.

However, T can also be positive when an intervention simply reduces the original separation between the two receivers:

$$2T_s = [M(x_{\mathrm{E}}) - M(x_{\mathrm{C}})] - [M(x_{\mathrm{E} \leftarrow \mathrm{C},s}) - M(x_{\mathrm{C} \leftarrow \mathrm{E},s})]. \tag{5}$$

For example, an intervention that makes both receivers' margins approach zero can yield positive T when their original margins favor their respective labels. Consequently, T measures a signed paired intervention effect, but does not by itself establish that the effect depends on whether the donor step is correct or erroneous.

Fixed-receiver donor contrast, D. To test this dependence, we compare correct and erroneous donors on the same receiver. For each receiver pair i, we select an external matched pair j from another problem. We then evaluate each receiver $r \in \{\mathrm{C}_i, \mathrm{E}_i\}$ twice at the same intervention setting: once with activations from the correct donor and once with activations from the erroneous donor. We define

$$D_{i,r,s} = M(x_{r \leftarrow \mathrm{E}_j,s}) - M(x_{r \leftarrow \mathrm{C}_j,s}), \qquad D_{i,\mathrm{common},s} = \tfrac{1}{2} \left(D_{i,\mathrm{C}_i,s} + D_{i,\mathrm{E}_i,s} \right). \tag{6}$$

Positive D means that, for the same receiver, the erroneous donor produces a more `INCORRECT`-favoring margin than the correct donor. Because the receiver is held fixed, its baseline margin cancels

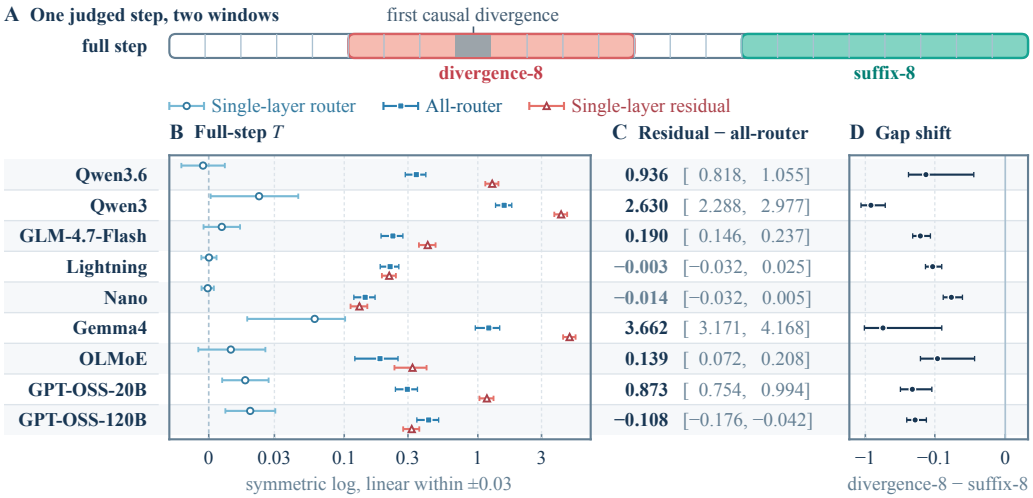

Figure 2: **Interface and coverage effects on the verification margin.** **A**: positions of the divergence-8 and suffix-8 windows within an evaluated step. **B**: full-step T for single-layer router, all-router, and residual interventions. **C**: the full-step residual-minus-all-router difference. **D**: the change in this difference from suffix-8 to divergence-8; negative values indicate a smaller residual-versus-routing gap near the first divergence. Error bars are pointwise 95% source-component percentile intervals from 50,000 resamples; direct contrasts use common eligible pairs. Panels **B** and **D** use symmetric-log axes.

from this comparison. This contrast tests differences between matched donor steps of different correctness; it does not isolate correctness from every other difference between their activations.

T and D therefore answer different questions. T summarizes signed changes from the two cross-role replacements within a correct/error pair, whereas D compares two donor roles on an unchanged receiver. Because they use different donor–receiver comparisons, neither D/T nor $T - D$ has a decomposition interpretation.

2.3 Interfaces, layer coverage, and token windows

Intervention interfaces. We replace activations at three points in an MoE block. A *router-logit* intervention replaces the scores assigned to candidate experts before expert selection. The model then uses its native expert selection, weighting, and dispatch procedures. A *MoE-output* intervention replaces the adapter-defined routed contribution before it is added to the residual stream. A *post-MoE residual* intervention replaces the resulting residual state, which can already contain information from attention, experts, and earlier layers.

Where an architecture includes a shared-expert or parallel dense branch, that branch remains receiver-derived under the MoE-output intervention (Section 3). Although all three interfaces are evaluated using the same verification margin, they replace different portions of the computation and should not be interpreted as additive stages of a single mediation decomposition.

Layer and token coverage. For local comparisons, we use one fixed, model-specific layer, referred to as the *anchor layer*. At this layer, we compare all three interfaces under four *token windows*, each specifying the positions within the evaluated reasoning step at which activations are replaced: the last token; the final eight tokens (*suffix-8*); eight tokens around the first token at which the paired prefixes diverge (*divergence-8*); and the full step.

We additionally apply a joint *all-router* intervention, replacing router logits at every routed layer under the same four token windows. Comparing suffix-8 with divergence-8 holds the number of intervened tokens fixed while changing their position. It does not, however, equalize the size of the activation changes or the computation preceding those positions. Exact window construction and donor–receiver token alignment are given in Appendix C.

We evaluate D both locally, at one layer and the final step token, and in a separate experiment that replaces router logits across all routed layers and the full step (Appendix F). We also evaluate an *all-layer MoE-output* intervention at the last token; this experiment is reported with T only.

2.4 Controls and uncertainty

Self-patches, which replace receiver activations with their own recorded values, and unpatched reruns verify that the intervention machinery preserves the receiver when no cross-example replacement is made. Matched external donors come from other problems, prioritizing agreement in error type and proximity of the error position, and remain fixed during analysis. Norm-random controls replace the donor–receiver activation difference with a Gaussian direction having the same ℓ_2 norm. For T, control donors preserve the donor role of each cross-role replacement; for D, both donor roles are evaluated on each fixed receiver.

Pairs receive equal weight. To account for dependence among pairs associated with the same source problem, we resample the $K = 235$ source-problem components and retain all pairs belonging to each sampled component. Last-token, full-step, D, and all-layer-output analyses use all $N = 290$ pairs. Eight-token and direct-contrast analyses use the corresponding common eligible subsets, with unavailable windows treated as missing rather than zero.

Primary intervals are two-sided 95% pointwise percentile intervals from 50,000 bootstrap resamples generated with PCG64. Direct differences are formed within pair before resampling, and reused matched donors are conditioned on rather than resampled independently. Exact seeds and version-specific analysis details are reported in Appendices C and F.

The checkpoints are fixed and the discovery sample was assembled retrospectively, so these intervals do not estimate a population distribution over models. We apply no multiplicity correction; post hoc receiver-role and GPT-checkpoint contrasts are reported as secondary analyses.

3 Models and Intervention Boundaries

We evaluate nine MoE checkpoints: Qwen3.6-35B-A3B, Qwen3-30B-A3B, GLM-4.7-Flash, Nemotron-3.5 Lightning 30B-A3B-BF16, Nemotron-3 Nano 30B-A3B, Gemma 4 26B-A4B, OLMoE-1B-7B, and GPT-OSS-20B/120B (Yang et al., 2025; Muennighoff et al., 2025; OpenAI, 2025; Z.ai, 2026).

Each checkpoint uses a fixed architecture-specific anchor layer for local interventions, while all-layer interventions cover every routed block in that checkpoint. Appendix B reports model sources, anchor indices, and routed-layer counts (Table 2). Qwen3 is the only checkpoint for which the local D and token-window experiments use different anchors: L36 and L24, respectively. Its probe protocol is described separately.

The MoE-output intervention follows each architecture's native computation boundaries. For GLM, Qwen3.6, and the Nemotron models, we replace the routed experts' contribution while retaining the receiver's shared-expert branch. For Gemma, we replace the normalized routed branch while retaining the receiver's parallel dense branch. For Qwen3, OLMoE, and GPT-OSS, the intervention replaces the native MoE/MLP-module output. Exact adapter definitions and execution checks are reported in Appendices B.3 and I, respectively.

4 Interface and Coverage Dependence on PRMBench

Before comparing interventions, we record the unpatched verification performance on this paired assay. Reading $M > 0$ as an `INCORRECT` verdict, accuracy ranges from 0.567 $[0.548, 0.588]$ in OLMoE to 0.786 $[0.755, 0.816]$ in Gemma and AUROC from 0.602 to 0.874, so every checkpoint is above the 0.5 chance level of this balanced assay while verification quality differs substantially (Appendix E).

4.1 Local residual interventions are stronger across token windows

Across nine models and four token windows, post-MoE residual interventions produce larger local responses (T) than both router-logit and MoE-output interventions: all 36 residual-minus-router and

all 36 residual-minus-output direct intervals are positive. This ordering therefore persists beyond the last-token setting and across different token regions within the evaluated step. The magnitude varies substantially across checkpoints; full-step residual T, for example, ranges from 0.1300 in Nano to 4.8698 in Gemma (Figure 2B). These values compare intervention interfaces within a checkpoint and should not be interpreted as model-quality rankings. Normalizing by the baseline margin gap of the same pairs, $R = \mathrm{mean}(T)/\mathrm{mean}(g)$ with $g = M(x_{\mathrm{E}}) - M(x_{\mathrm{C}})$, places full-step residual effects between 0.24 $[0.22, 0.26]$ in GPT-OSS-120B and 0.90 $[0.88, 0.93]$ in GPT-OSS-20B; Gemma, which has the largest absolute T, has $R = 0.69$ $[0.65, 0.72]$. In GPT-OSS-20B, Qwen3, Qwen3.6, and Gemma the R interval lies entirely above 0.5, so the intervention does not merely compress the paired gap in Equation 5 but reverses its sign on average. These four checkpoints have patched accuracy of 0.27 to 0.37 after this intervention (Appendix E).

4.2 Local residual effects depend on donor correctness

We next ask whether the intervention direction depends on whether the replacement activation comes from a correct or erroneous step. At one layer and the last token, residual D is positive on both correct and error receivers in eight of nine models (Table 1). Residual-minus-router and residual-minus-output contrasts in D_{common} are also positive in these models. By comparison, MoE-output D is positive on both receiver types in five models, while every single-router D_{common} interval includes zero. Thus, the local interface difference extends beyond response magnitude: correctness-dependent effects are much clearer at the residual interface than at the router-logit interface.

Table 1: Fixed-receiver D at the last token. Positive values mean that, on the same receiver, activations from an erroneous step produce a more `INCORRECT`-favoring margin than activations from a correct step. Residual D is positive on both receiver types in eight models; Qwen3.6 (†) is the exception. Router and output columns report D_{common}, while residual columns retain the two receiver types. All panels use $N = 290$ pairs and $K = 235$ source components with pointwise 95% percentile intervals.

Model	Residual D_{correct}	Residual D_{error}	Router D_{common}	Output D_{common}
Qwen3.6†	−.0096 [−.0308,.0130]	.0356 [.0101,.0625]	−.0010 [−.0085,.0062]	−.0053 [−.0120,.0010]
Qwen3	.7282 [.6313,.8269]	.5921 [.4989,.6879]	−.0073 [−.0151,.0006]	.0519 [.0339,.0698]
GLM-4.7-Flash	.1989 [.1594,.2404]	.1371 [.1077,.1673]	.0009 [−.0062,.0080]	.0642 [.0475,.0806]
Lightning	.1492 [.1285,.1704]	.1125 [.0953,.1304]	−.0001 [−.0035,.0034]	−.0003 [−.0041,.0036]
Nano	.0793 [.0680,.0907]	.0624 [.0525,.0726]	.0005 [−.0017,.0026]	.0004 [−.0024,.0032]
Gemma4	1.1705 [.8775,1.4791]	.2586 [.1150,.4163]	−.0061 [−.0208,.0087]	.1188 [.0681,.1793]
OLMoE	.1368 [.0859,.1925]	.1586 [.1049,.2169]	.0050 [−.0011,.0110]	.0349 [.0211,.0487]
GPT-OSS-20B	.1352 [.1016,.1696]	.1019 [.0726,.1323]	−.0017 [−.0060,.0027]	−.0014 [−.0108,.0079]
GPT-OSS-120B	.2262 [.1895,.2645]	.1654 [.1342,.1977]	.0061 [−.0025,.0150]	.0436 [.0312,.0564]

Qwen3.6 shows clear receiver dependence. Its residual D is -0.0096 $[-0.0308, 0.0130]$ on correct receivers and 0.0356 $[0.0101, 0.0625]$ on error receivers, with a direct difference of -0.0452 $[-0.0748, -0.0156]$. Its $D_{\mathrm{common}} = 0.0130$ $[-0.0054, 0.0327]$ therefore does not establish a common positive direction. Additional receiver asymmetries are reported in Appendix D.

4.3 Broader routing coverage changes the interface ordering

Single-layer router interventions are weak, but this does not bound the effect of intervening on routing more broadly. Full-step all-router T is positive in all nine models and increases over the corresponding single-router interventions (Figure 2B). At full-step coverage, the residual-minus-all-router gap remains positive in six models, is unresolved in Lightning and Nano, and reverses in GPT-OSS-120B (-0.1083 $[-0.1757, -0.0419]$; Figure 2C). At divergence-8, all-router also exceeds residual in both Nemotron checkpoints and GPT-OSS-120B. Routing can therefore become comparable to, or stronger than, a local residual intervention when coverage expands across layers and tokens.

Increasing token coverage alone does not produce a common trend in the residual-versus-routing gap. A clearer pattern appears when token count is held fixed. Define $G_w = T_{\mathrm{res},w} - T_{\text{all-router},w}$. Moving from suffix-8 to divergence-8 reduces G_w in all nine models, with every direct interval

for $G_{\text{divergence8}} - G_{\text{suffix8}}$ below zero (Figure 2D). Thus, the relative effect of routing depends systematically on token position, not only on the number of tokens intervened on.

This positional effect does not imply that the absolute routing response is always larger near divergence. For GPT-OSS-120B, the all-router divergence-minus-suffix difference is -0.0028 $[-0.0591, 0.0535]$, whereas the residual difference is -0.1946 $[-0.2315, -0.1589]$. Here, the smaller residual-versus-routing gap near divergence is driven primarily by residual sensitivity to position.

The two GPT-OSS checkpoints further separate intervention magnitude from the correctness dependence tested below. GPT-OSS-120B has the larger full-step all-router T ($120\text{B} - 20\text{B} = 0.1316$ $[0.0704, 0.1935]$), motivating a direct test of whether it also shows the stronger fixed-receiver effect.

4.4 Broad routing correctness dependence varies across models and receivers

We extend the fixed-receiver analysis across all routed layers and the full evaluated step. Under this broader routing intervention, correctness-dependent effects are substantially less uniform than the local residual effects above. GPT-OSS-20B has positive $D_{\text{common}} = 0.0699$ $[0.0181, 0.1226]$ (Figure 3), with $D_{\text{correct}} = 0.1137$ $[0.0480, 0.1803]$ and $D_{\text{error}} = 0.0262$ $[-0.0335, 0.0871]$. Gemma also shows a positive correct-receiver effect, 0.3875 $[0.0762, 0.7002]$, although its $D_{\text{common}} = 0.2207$ $[-0.0114, 0.4545]$ remains unresolved. The other checkpoints likewise do not establish a common positive effect.

Direct contrasts confirm that response magnitude and correctness dependence can order checkpoints differently. In GPT-OSS-20B, $D_{\text{correct}} - D_{\text{error}} = 0.0874$ $[0.0158, 0.1597]$, indicating a larger effect on correct receivers. More importantly, the paired GPT-OSS-20B-minus-120B contrast in D_{common} is 0.0751 $[0.0206, 0.1301]$, while GPT-OSS-120B itself has $D_{\text{common}} = -0.0051$ $[-0.0393, 0.0290]$. Thus, GPT-OSS-120B has the larger broad-routing T, but GPT-OSS-20B has the larger fixed-receiver D. A larger routing-induced judgment change therefore does not necessarily imply a stronger dependence on the correctness of the replacement activation.

These checkpoint contrasts do not isolate model scale: training, routed depth, expert count, and checkpoint identity also differ. Likewise, local residual D and broad router D use different layer and token coverage and therefore do not provide a matched-coverage interface comparison. Additional scoring sensitivities and post hoc contrasts are reported in Appendix F.

5 All-Layer Output Interventions and Donor Controls

Single-layer MoE-output interventions can be weak, so we next test whether replacing MoE outputs jointly across all routed layers produces a stronger response. At the last token, the all-layer MoE-output intervention has positive T in all nine checkpoints, ranging from 0.0994 in Qwen3.6 to 1.1326 in Gemma (Table 6). Moreover, every paired-donor-minus-norm-random interval is positive. Thus, the effect of the all-layer output intervention cannot be explained by activation displacement magnitude alone.

Matched external donors from other problems also produce substantial responses. All paired-donor-minus-matched-donor intervals include zero, so the current data do not resolve an advantage for the paired donor over a matched external donor; nor do they establish equivalence. The result instead shows that large MoE-output intervention effects can also arise from donor activations drawn from other problems with matched error type and boundary position.

We apply the same controls to the broader router interventions. Full-step all-router T exceeds the norm-random control in all nine checkpoints, although the contrast is close to zero for OLMoE (0.0466 $[0.0003, 0.0938]$). By contrast, matched external donors produce larger T than the paired donors in eight checkpoints, with OLMoE unresolved. Because matched donors can differ in activation magnitude, context, and structure, this comparison should not be interpreted as evidence that they contain more correctness-related information. Full control results are reported in Appendix D.

A decision-level reading gives the same picture. The full-step all-router intervention changes the verdict of 0.06 to 0.23 of receivers under the paired donor, whereas the norm-random control changes

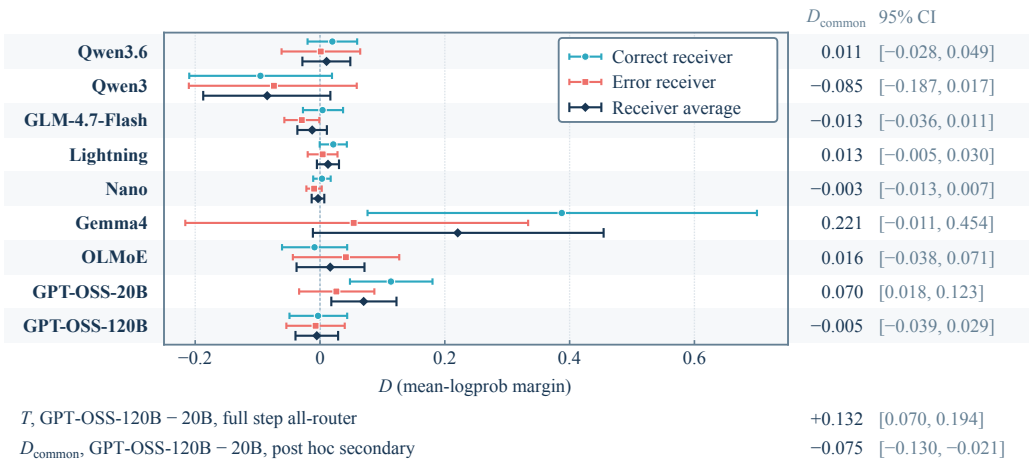

Figure 3: **Fixed-receiver effects for full-step all-router interventions.** Positive D means that, on the same receiver, activations from an erroneous step produce a more `INCORRECT`-favoring margin than activations from a correct step. Circles and squares show the two receiver types, and diamonds show their average D_{common}. Error bars are pointwise 95% source-component percentile intervals ($N = 290$, $K = 235$, 50,000 resamples). The lower strip reports direct GPT-OSS checkpoint contrasts for T and D_{common}.

0.03 to 0.11 and the matched external donor 0.07 to 0.35, exceeding the paired donor in all nine checkpoints; a flip rate on its own therefore does not establish donor or routing specificity (Appendix E).

These controls distinguish intervention magnitude from two stronger claims. Exceeding the norm-random control shows that the observed response is not explained by displacement norm alone, while the matched-donor comparison does not establish pair-specificity. Neither comparison determines whether the judgment change follows the correctness of the replacement activation; that question requires the fixed-receiver D analysis in Section 4.4. The all-layer MoE-output intervention currently has no corresponding fixed-receiver D experiment. Because the output, router, and residual experiments also differ in interface and coverage, they should not be interpreted as a router-to-output-to-residual mediation decomposition.

6 Representational Signals and Natural-Error Interventions

Using out-of-fold L2-logistic probes grouped by source problem, router-logit representations distinguish correct from erroneous steps across eight evaluated checkpoints. A readable signal does not by itself determine intervention strength (Appendix G).

The interface difference also appears on naturally occurring errors: residual interventions produce larger absolute changes than router-logit interventions in all six model-by-direction comparisons. This exploratory ProcessBench analysis uses GPT-OSS-20B, Gemma, and Nemotron-3 Nano with 64 cross-problem pairs; it does not include fixed-receiver D or token-window analyses (Appendix H).

7 Related Work

Error signals and step-level verification. Process supervision and fine-grained reasoning benchmarks motivate step-level verification (Lightman et al., 2024; Song et al., 2025; Zheng et al., 2025). Prior work shows that internal representations can contain information about truthfulness, reasoning correctness, or answerability that is not directly reflected in model outputs (Orgad et al., 2025; Yuan et al., 2026; Du & Hu, 2026), and intermediate-state probes have been used for self-verification and early exit (Zhang et al., 2025). Yuan et al. also report limited success in improving final-answer accuracy; we measure verification-margin changes on fixed step prefixes. Other work directly intervenes on verifier representations, for example by steering verifier strictness (Zhou et al., 2026). Our work connects these two lines by asking not only whether error-related information is present, but

also when changing router, MoE-output, or residual activations alters the verification judgment and whether that change depends on the correctness of the replacement activation.

MoE routing and intervention. MoE routing reflects semantic structure (Olson et al., 2025) and hidden-state geometry (Wang et al., 2026), while other studies analyze expert specialization, the upstream formation of routing decisions, and experts that are important for generation (Herbst et al., 2026; Li et al., 2026; Su et al., 2026). Intervening on experts or routing can also change model behavior (Fayyaz et al., 2026; Bandarkar et al., 2026), and recent work has begun to separate routing diagnostics from behavioral influence under ablation, quantization, or controlled gate changes (Engmann et al., 2026; Gu, 2026; Caruzzo et al., 2026). Related analyses study router-visible subspaces, expert allocation, factual recall, and expert selectivity (Ye et al., 2026; Yoon et al., 2026; Lu et al., 2026; Salomone et al., 2026). We complement these studies with a controlled comparison of router-logit, MoE-output, and post-MoE residual interventions, varying both layer coverage and token position under a common step-level verification endpoint.

Interpreting activation replacement. Causal tracing and Patchscopes study model behavior through activation or representation replacement (Vig et al., 2020; Meng et al., 2022; Ghandeharioun et al., 2024). A central limitation of such analyses is that information readable from a representation need not be causally used (Hewitt & Liang, 2019; Elazar et al., 2021; Sharma et al., 2026), while patching conclusions can depend on the intervention construction, evaluation metric, and the layers and token positions patched (Zhang & Nanda, 2024). Subspace interventions and causal localization introduce further interpretational limitations (Makelov et al., 2024; Hase et al., 2023). Following these concerns, we separately measure the size of an intervention-induced judgment change and, with the receiver held fixed, whether the direction of that change depends on whether the replacement activation comes from a correct or erroneous step. We do not interpret either quantity as a pathway contribution or causal fraction.

8 Limitations

Activation replacement can move model states off their natural distribution, and its effects depend on the intervention site, evaluation margin, and the amount of computation being replaced (Zhang & Nanda, 2024; Makelov et al., 2024). Moreover, a positive fixed-receiver D shows that the intervention effect depends on whether the replacement activation comes from a correct or erroneous step, but it does not isolate correctness from correlated differences in text or representation geometry. Finally, our nine checkpoints form a fixed, retrospectively assembled set rather than a sample from a model population. Accordingly, our results characterize the tested interventions and settings; they do not identify unique computational pathways or causal fractions of the model's verification process.

9 Conclusion

We introduced RouteVerify to study when and where routing interventions affect step-level verification in MoE models. Across nine checkpoints and four token windows, local post-MoE residual interventions consistently produce larger judgment changes than local router-logit or MoE-output interventions. However, routing interventions become substantially stronger when expanded across routed layers and tokens, and equal-length windows reveal systematic dependence on token position.

More importantly, intervention magnitude and correctness-dependent change are not the same quantity. Fixed-receiver analyses show clear correctness-dependent effects for local residual interventions in most checkpoints, but substantially more variation for broad routing interventions; the two GPT-OSS checkpoints even reverse their ordering under T and D. All-layer MoE-output and routing controls further show effects beyond norm-matched random perturbations without resolving an advantage for the paired donor over matched external donors.

Together, these results show that routing cannot be characterized by a single intervention effect. Its impact on step-level verification depends on the intervention interface, layer and token coverage, and token position, while information readable from a representation, the size of a judgment change, and whether that change follows the correctness of the replacement activation should be evaluated separately.

REPRODUCIBILITY STATEMENT

Experiments record frozen dataset and model revisions, pair hashes, resolved configurations, seeds, commits, validation-access flags, and immutable predictions. Appendix J reports anonymous run IDs and hashes; the method and control appendices specify estimands, paired resampling, identity checks, and protocol controls. Appendix F separates the wide-D primary analysis, total-logprob sensitivity, and post hoc direct contrasts, including their input-identity and recovery records. Code, configurations, pair manifests, and non-weight artifacts are available for review at `https://anonymous.4open.science/r/RouteVerify-A370/`. Recorded revisions identify separately downloadable model weights.

AI USE STATEMENT

Generative AI tools, including OpenAI Codex, Claude Code, and Cursor, assisted with research framing and experimental design; implementation, testing, data selection and reformatting, analysis, and interpretation; and literature review, figure preparation, translation, and manuscript drafting and editing. They were not used to generate study datasets or proofs, and external audit outputs were treated as issue leads rather than evidence. AI-assisted outputs were checked against source code, tests, machine-readable artifacts, and cited sources; no AI system had authority to accept results or determine the scientific conclusions. The authors take responsibility for the final text, claims, code, and artifacts.

REFERENCES

Lucas Bandarkar, Chenyuan Yang, Mohsen Fayyaz, Junlin Hu, and Nanyun Peng. Multilingual routing in mixture-of-experts. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/1b558190825286a3defcc78d02fa2189-Paper-Conference.pdf.

Cedric Caruzzo, Donggeun Yoo, and Tae Soo Kim. Routing divergence is not evidence of behavioral influence in same-weight MoE self-distillation, 2026. URL https://arxiv.org/abs/2608.15787.

Yucheng Du and Xiyang Hu. Recognition-refusal misalignment in LLMs: Why models answer structurally unanswerable questions, 2026. URL https://arxiv.org/abs/2608.29109. Accepted to EMNLP 2026 Main Conference.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021. doi: 10.1162/tacl_a_00359. URL https://aclanthology.org/2021.tacl-1.10/.

Leonard Engmann, Christian Medeiros Adriano, and Holger Giese. From observation to intervention: A causal audit of expert importance in mixture-of-experts models, 2026. URL https://arxiv.org/abs/2606.10703. Accepted at the ICML 2026 Workshop on Philosophy of Science Meets Machine Learning (PhilML), non-archival.

Mohsen Fayyaz, Seyed MohammadAli Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Ryan Rossi, Trung Bui, Hinrich Schuetze, and Nanyun (Violet) Peng. Steering MoE LLMs via expert (de)activation. In *International Conference on Learning Representations*, pp. 7803–7826, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/0d61c5f5ef91e7e8a091b7b8f72b853c-Paper-Conference.pdf.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15466–15490. PMLR, 2024. URL https://proceedings.mlr.press/v235/ghandeharioun24a.html.

Parvel Gu. Detecting a route flip is easier than knowing whether to fix it: Causal route-mediated damage in quantized mixture-of-experts, 2026. URL https://arxiv.org/abs/2608.11212.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 17643–17668. Curran Associates, Inc., 2023. doi: 10.52202/075280-0774. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/3927bbdcf0e8d1fa8aa23c26f358a281-Paper-Conference.pdf.

Jeremy Herbst, Stefan Wermter, and Jae Hee Lee. The expert strikes back: Interpreting mixture-of-experts language models at expert level. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2604.02178.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

Wengang Li, Lingqi Zhang, Toshio Endo, and Mohamed Wahib. Understanding cross-layer contributions to mixture-of-experts routing in LLMs. In *International Conference on Learning Representations*, pp. 37420–37446, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/3efdd3cd3d70da09ec6ed3b3d17d2fd5-Paper-Conference.pdf.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/aca97732e30bcf1303bc22ac3924fd16-Paper-Conference.pdf.

Yuetian Lu, Ali Modarressi, Yihong Liu, and Hinrich Schütze. Expert-aware causal tracing of factual recall in sparse MoE language models, 2026. URL https://arxiv.org/abs/2606.03780v2. Version 2, revised September 3, 2026.

Aleksandar Makelov, Georg Lange, Atticus Geiger, and Neel Nanda. Is this the subspace you are looking for? an interpretability illusion for subspace activation patching. In *International Conference on Learning Representations*, volume 2024, pp. 26486–26515, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/70b8505ac79e3e131756f793cd80eb8d-Paper-Conference.pdf.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372. Curran Associates, Inc., 2022. doi: 10.52202/068431-1262. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/6f1d43d5a82a37e89b0665b33bf3a182-Paper-Conference.pdf.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah Smith, Pang Wei Koh, Amanpreet Singh, and Hanna Hajishirzi. OLMoE: Open mixture-of-experts language models. In *International Conference on Learning Representations*, pp. 62061–62121, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/9b224ace8963c9385ad5e2b5c9039b97-Paper-Conference.pdf.

Matthew Lyle Olson, Neale Ratzlaff, Musashi Hinck, Man Luo, Sungduk Yu, Chendi Xue, and Vasudev Lal. Probing semantic routing in large mixture-of-expert models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 18263–18278. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.991. URL https://aclanthology.org/2025.findings-emnlp.991/.

OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. In *International Conference on Learning Representations*, pp. 66880–66913, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a712d461e57201efe35d429a6f1731c1-Paper-Conference.pdf.

Tony Salomone, Deep Gandhi, and Ali Asaria. How modular is a frontier mixture-of-experts? a pre-registered causal test in which apparent expert modularity mostly dissolves, 2026. URL https://arxiv.org/abs/2606.25092.

Aryan Sharma, Cutter Dawes, and Shivam Raval. Dissociating decodability and causal use in bracket-sequence transformers, 2026. URL https://arxiv.org/abs/2604.22128.

Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. PRMBench: A fine-grained and challenging benchmark for process-level reward models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25299–25346. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1230. URL https://aclanthology.org/2025.acl-long.1230/.

Zunhai Su, Qingyuan Li, Hao Zhang, Weihao Ye, Qibo Xue, Yulei Qian, Yuchen Xie, Ngai Wong, and Kehong Yuan. Unveiling super experts in mixture-of-experts large language models. In *International Conference on Learning Representations*, pp. 118323–118354, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/c01b90b643d0e877cff0f34cbe27410f-Paper-Conference.pdf.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12388–12401. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/92650b2e92217715fe312e6fa7b90d82-Paper.pdf.

Xi Wang, Soufiane Hayou, and Eric Nalisnick. The myth of expert specialization in MoEs: Why routing reflects geometry, not necessarily domain expertise, 2026. URL https://arxiv.org/abs/2604.09780.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Charles Ye, Bo Yuan, and Lee Sharkey. Polysemantic experts, monosemantic paths: Routing as control in MoEs, 2026. URL https://arxiv.org/abs/2604.17837.

Youngsik Yoon, Siwei Wang, Wei Chen, and Jungseul Ok. When are experts misrouted? counterfactual routing analysis in mixture-of-experts language models, 2026. URL https://arxiv.org/abs/2605.07260.

Aojie Yuan, Zhiyuan Julian Su, Haiyue Zhang, Yi Nian, and Yue Zhao. Hidden error awareness in chain-of-thought reasoning: The signal is diagnostic, not causal, 2026. URL https://arxiv.org/abs/2605.09502.

Z.ai. GLM-4.7-Flash model card. Hugging Face, 2026. URL https://huggingface.co/zai-org/GLM-4.7-Flash.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reasoning models know when they're right: Probing hidden states for self-verification. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=O6I0Av7683.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, volume 2024, pp. 1651–1678, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/06a52a54c8ee03cd86771136bc91eb1f-Paper-Conference.pdf.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. ProcessBench: Identifying process errors in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1009–1024. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.50. URL https://aclanthology.org/2025.acl-long.50/.

Yefan Zhou, Yilun Zhou, Austin Xu, Soroush Vosoughi, Shafiq Joty, and Jiang Gui. The hidden signal of verifier strictness: Controlling and improving step-wise verification via selective latent steering, 2026. URL https://arxiv.org/abs/2605.20745.

A Appendix Roadmap

The appendix provides implementation details, supplementary analyses, controls, and reproducibility records for the experiments in the main paper. It is organized by scientific function rather than by experiment chronology:

1. Appendix B: verification setup, model identifiers, intervention layers, and architecture-specific boundaries;
2. Appendix C: fixed-receiver contrasts, token windows, donor construction, controls, and statistical analysis;
3. Appendix D: supplementary nine-checkpoint results and control comparisons;
4. Appendix E: unpatched verification performance, decision flip rates, and gap-normalized effects;
5. Appendix F: full-step all-router fixed-receiver analysis, scoring sensitivity, and direct contrasts;
6. Appendix G: representation-probe analyses;
7. Appendix H: natural-error evaluation on ProcessBench;
8. Appendix I: additional implementation checks and control interpretation;
9. Appendix J: reproducibility and artifact records; and
10. Appendix K: claim boundaries.

B Model and Verification Setup

B.1 Prompt and verification endpoint

The verification prompt contains the mathematical problem, the solution trace through the evaluated step, and a fixed request to choose between `CORRECT` and `INCORRECT`. Dataset labels, pair identities, generator metadata, and later solution steps are not shown to the model. The scorer computes the mean conditional log probability of each complete candidate string using the model's native tokenizer and chat template. Intervention runs retain the receiver prompt and modify only the designated internal activation.

PRMBench pairs share a problem and an identical prefix before the first benchmark-annotated error step. The nine-checkpoint intervention analyses use 290 ordered pairs grouped into 235 canonical source-problem components. ProcessBench is analyzed separately with 64 cross-problem pairs. Each archived analysis records its sample count, pair identity, bootstrap identity, and source-prediction path.

We use the frozen `hitsmy/PRMBench_Preview` snapshot recorded in the release manifests. Eligible pairs have matching nonempty problems, equal nonzero step counts, valid error indices with an unambiguous indexing base, identical text before the first annotated error, and a changed step at that boundary. Only complete, unique pairs are retained. An error-type-stratified 50/50

split of 580 complete pair groups (seed 20260730) assigns 290 to discovery and 290 to validation. The nine-checkpoint intervention experiments use only the discovery partition. The 235 source components are bootstrap units, not the units of this split. Labels follow the benchmark annotations without independent mathematical review of every trace and error boundary.

Category-exclusion sensitivity. In a post hoc analysis, we exclude the six circular and three redundancy pairs in the Simplicity category, retaining the original predictions, donor assignments, and 50,000-resample source-component bootstrap protocol (seed 20260908). Full-step and last-token panels retain 281 pairs in 230 components; eight-token contrasts use their common eligible subsets. All 72 single-layer residual-minus-router and residual-minus-MoE-output T intervals remain positive, and all nine divergence-8-minus-suffix-8 gap intervals remain negative. For OLMoE, the full-step all-router paired-minus-norm-random contrast changes from 0.0466 $[0.0003, 0.0938]$ to 0.0450 $[-0.0013, 0.0928]$: the point estimate changes little, but the interval includes zero. The full sample remains the primary analysis.

B.2 Models and intervention layers

The nine evaluated checkpoints are Qwen3.6-35B-A3B, Qwen3-30B-A3B, GLM-4.7-Flash, Nemotron-3.5 Lightning 30B-A3B-BF16, Nemotron-3 Nano 30B-A3B, Gemma 4 26B-A4B, OLMoE-1B-7B, and GPT-OSS-20B/120B. Model identifiers and source links follow.

`google/gemma-4-26B-A4B-it`,
`nvidia/NVIDIA-Nemotron-3-Nano-30B-A3B-BF16`,
`nvidia/NVIDIA-Nemotron-3.5-Lightning-30B-A3B-BF16`,
`Qwen/Qwen3.6-35B-A3B`, `Qwen/Qwen3-30B-A3B`, `zai-org/GLM-4.7-Flash`,
`allenai/OLMoE-1B-7B-0924-Instruct`, `openai/gpt-oss-20b`, and
`openai/gpt-oss-120b`. Exact model and tokenizer revisions are retained in the resolved configurations and artifact index.

Table 2: Layer coverage in the nine-checkpoint analysis. Layers are zero-indexed absolute decoder-block indices. "All routed sites" counts routed blocks replaced by an all-router or all-layer MoE-output intervention, not every decoder block. "Probe layer" refers only to completed representation analyses in Appendix G.

Model	Local D layer	Window layer	All routed sites	Probe layer
Qwen3.6-35B-A3B	20	20	40	20
Qwen3-30B-A3B	36	24	48	24
GLM-4.7-Flash	24	24	46	24
Nemotron-3.5 Lightning	27	27	23	27
Nemotron-3 Nano	27	27	23	27
Gemma 4 26B-A4B	15	15	30	15
OLMoE-1B-7B	8	8	16	—
GPT-OSS 20B	12	12	24	12
GPT-OSS 120B	18	18	36	18

B.3 Architecture-specific intervention boundaries

Router-logit interventions replace the adapter's pre-dispatch logits while retaining native expert selection and weighting. MoE-output interventions replace the routed contribution at the architecture-specific output boundary, while post-MoE residual interventions replace the corresponding residual activation. These are distinct intervention locations and are not treated as additive stages of a mediation decomposition.

For Qwen3.6 and GLM, the MoE-output boundary precedes addition of the shared branch, so the shared-expert contribution remains receiver-derived. GLM has 47 decoder blocks, with dense block 0 excluded from the 46 routed sites. In both Nemotron checkpoints, the L27 output boundary is the final routed-only aggregate at `fc2_latent_proj`; the receiver shared-expert branch remains unchanged, and the 23 routed expert blocks lie within 52 hybrid decoder blocks.

Gemma replaces the normalized routed contribution at `post_feedforward_layernorm_2` while preserving its parallel dense branch. Its residual boundary is post-FFN. OLMoE exposes the routed aggregate before residual addition. Qwen3 and the GPT-OSS checkpoints expose the native MoE/MLP-module output. GPT-OSS uses 24 and 36 routed blocks for the 20B and 120B checkpoints, respectively, under native MXFP4 execution and routing rules. The executed adapters and resolved configurations define the exact intervention boundaries.

C Intervention Protocol Details

C.1 Single-layer fixed-receiver correctness contrast

For receiver pair i, we select an external matched donor pair $d(i)$ from a different canonical source component. Matching prioritizes agreement in error type, followed by proximity of the first-error position, with a seeded pair hash used for tie-breaking. The same external pair supplies its correct and error activations to both receivers. Donor assignments remain fixed during analysis; the 290 receiver pairs use 204 distinct external donor pairs.

Let $M_i^{r \leftarrow d,s}$ denote the verification margin after replacing the designated last-token receiver activation with the corresponding activation from donor role s, for receiver role $r \in \{\mathrm{C}, \mathrm{E}\}$. The four donor/receiver combinations give

$$D_{i,r} = M_i^{r \leftarrow d,\mathrm{E}} - M_i^{r \leftarrow d,\mathrm{C}}, \tag{7}$$

$$D_{i,\mathrm{common}} = \tfrac{1}{2}(D_{i,\mathrm{C}} + D_{i,\mathrm{E}}). \tag{8}$$

Because the receiver is identical in the two terms defining $D_{i,r}$, its baseline verification margin cancels. Receiver roles are averaged within pair before computing D_{common}, so the sample size remains 290 rather than 580. Direct interface contrasts are likewise computed within pair before resampling.

Positive D means that, for the same receiver, replacing the activation with one from an erroneous step produces a more `INCORRECT`-favoring margin than replacing it with one from a correct step. It therefore rules out a completely donor-role-independent perturbation as a sufficient explanation for that comparison. These labels remain correlated with text, content, and activation geometry, however. Neither D/T nor $T - D$ has a decomposition interpretation. This local experiment uses one layer and the final step token; the full-step all-router fixed-receiver analysis is reported in Appendix F.

C.2 Token windows and alignment

The window experiment crosses four intervention conditions—single-layer router logits, single-layer MoE output, single-layer residual, and all-layer router logits—with four token regions: the final token of the evaluated step, the final k tokens, k tokens centered on the first token-level divergence between the paired rendered prefixes, and the full evaluated step.

For each tokenizer, we choose the largest $k \in \{4, 8, 16, 32\}$ for which both paired steps contain at least k tokens in at least 90% of discovery pairs, using token lengths only. All nine checkpoints select $k = 8$. Divergence is defined as the first difference in the complete rendered causal prefixes. The centered window is clipped to remain within the evaluated step without shortening it. If the first divergence lies outside that step, the example is ineligible for this condition. For even k, the divergence token occupies the left of the two central positions before clipping. Suffix and divergence windows may overlap or coincide.

Full-step replacement covers every receiver-step token even when the donor step has a different length. If receiver and donor windows have lengths R and L, receiver offset j maps to donor offset

$$\lfloor j(L-1)/(R-1) + 1/2 \rfloor ;$$

when $R = 1$, the final donor token is used. The mapping preserves endpoint ordering and may reuse donor positions. It is an operational alignment rather than a claim of semantic token equivalence.

C.3 Control conditions

Each interface/window combination includes paired-donor, self-patch, matched-external-donor, and norm-random controls. Self and matched conditions inherit the paired receiver positions. A matched

donor supplies its own corresponding window; unavailable donor windows are recorded as missing. Norm-random interventions use a Gaussian direction whose ℓ_2 displacement matches the paired donor–receiver displacement separately at each intervention position and layer. Equal token count therefore does not imply equal activation displacement or equal cumulative layer coverage.

A pair contributes to T only when both replacement directions are available. Direct control and cross-window contrasts use the intersection of their eligible pair sets, and unavailable windows are never treated as zero. Table 3 reports the resulting sample sizes. Last-token and full-step panels use all $N = 290$ pairs and $K = 235$ source components.

Table 3: Eligible pairs/source components (N/K) for the eight-token windows. Paired-donor, self-patch, and norm-random conditions share the paired mask. "Common" is the suffix/divergence intersection used for the equal-length position contrast. Matched-donor comparisons use their own complete-pair intersection.

	Suffix-8		Divergence-8		Common
Model	Paired	Matched	Paired	Matched	Paired
Qwen3.6	278/226	265/217	276/224	261/213	276/224
Qwen3	278/226	265/217	276/224	261/213	276/224
GLM-4.7-Flash	274/222	256/209	272/220	252/205	272/220
Lightning	278/226	265/217	276/224	261/213	276/224
Nano	278/226	265/217	276/224	261/213	276/224
Gemma 4	279/226	268/220	277/224	264/216	277/224
OLMoE	271/220	250/205	271/220	250/205	271/220
GPT-OSS 20B	272/220	253/206	271/219	251/204	271/219
GPT-OSS 120B	272/220	253/206	271/219	251/204	271/219

C.4 All-layer MoE-output intervention

The all-layer MoE-output experiment replaces the last-token routed output at every routed site listed in Table 2. It includes paired donors, matched external donors, and norm-random controls. The norm-random condition matches the paired donor–receiver displacement independently at each routed site. Paired-minus-matched and paired-minus-random contrasts are computed within pair before aggregation. This intervention differs in both interface and coverage from a single-layer MoE-output intervention and a full-step all-router intervention.

C.5 Statistical analysis

The nine-checkpoint window analysis reconstructs 161,480 pair-level T values from the saved scores and includes all four token windows, intervention interfaces, controls, direct interface comparisons, cross-window contrasts, and gap-change contrasts. Uncertainty is estimated by 50,000 PCG64 source-component bootstrap resamples. Each draw samples eligible source components with replacement, retains all pairs belonging to each sampled component, and computes a pair-weighted mean. Analyses sharing the same eligibility mask use the same resamples.

Reported intervals are two-sided pointwise 95% percentile intervals. Inference conditions on the fixed checkpoints, discovery pool, and donor assignments; reused donors are not independently resampled. No multiplicity correction or population-of-models test is applied.

All final single-layer D, all-layer MoE-output, window, and direct-contrast panels use 50,000 source-component resamples. Exact seeds and analysis identifiers are stored with the archived analysis artifacts. The reported window figures use one common analysis throughout, and direct differences are formed within pair before resampling.

D Supplementary Nine-Checkpoint Results

Receiver dependence. Qwen3.6 is the only checkpoint for which the local residual intervention does not show a common positive direction across receiver types. $D_{\mathrm{C}} = -0.00964$

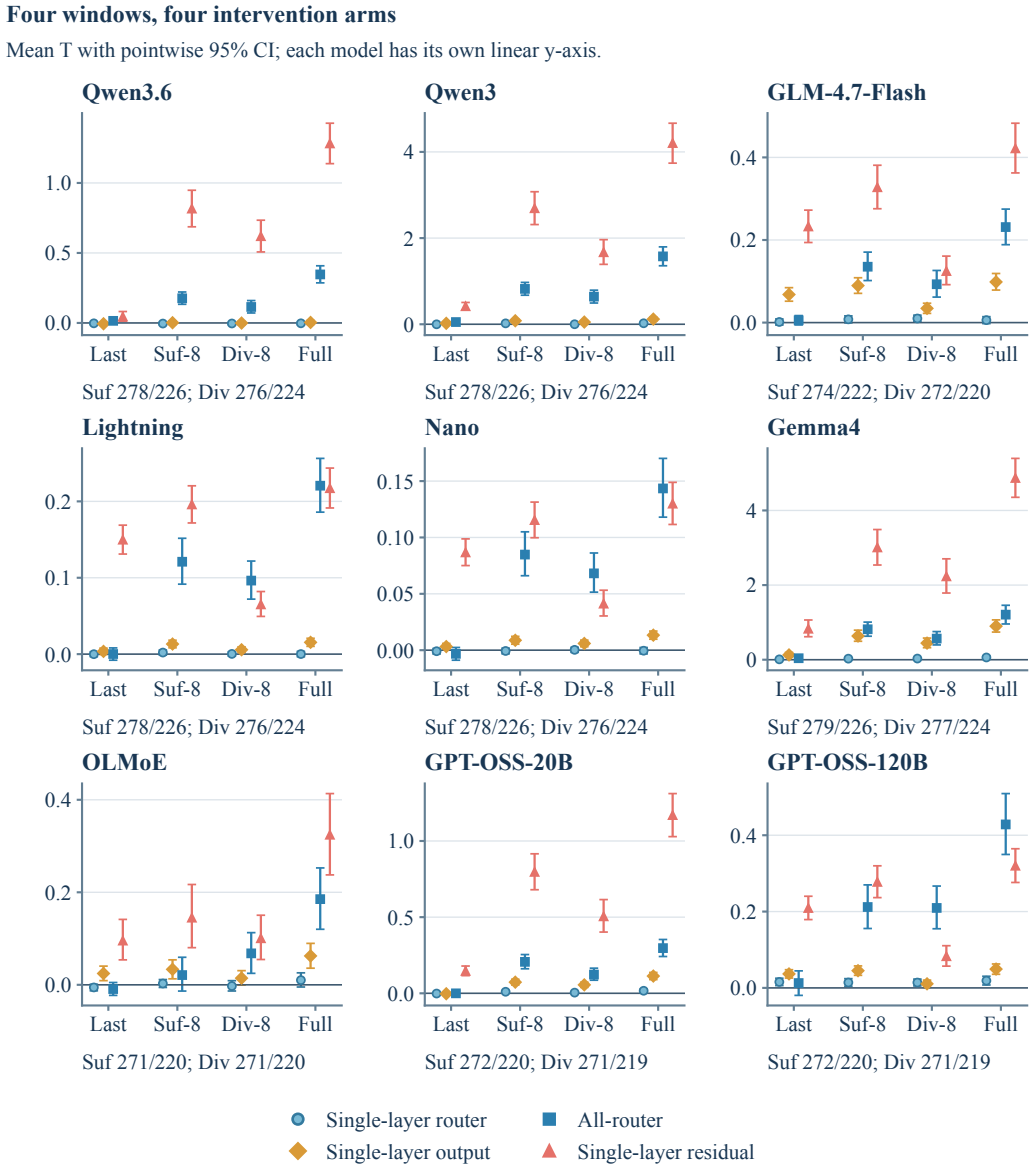

Figure 4: Nine-checkpoint intervention profiles across four token windows and four intervention conditions. Panels use checkpoint-specific linear axes. Points are pair-weighted T means with two-sided pointwise 95% source-component percentile intervals from 50,000 PCG64 resamples. Last-token and full-step conditions use $N = 290$, $K = 235$; eight-token sample sizes are given in Table 3. Local intervention layers are fixed within each checkpoint, whereas the all-router condition replaces router logits at every routed layer.

Table 4: Last-token fixed-receiver D at all three local interfaces. Each checkpoint uses $N = 290$ receiver pairs and $K = 235$ source components. D_{common} averages the two receiver types within pair. Intervals are two-sided pointwise 95% source-component percentile intervals from 50,000 resamples. Qwen3 uses L36 for this experiment and L24 for the token-window experiment.

Model	D_{correct}	D_{error}	D_{common}
Router			
Qwen3.6	.0015 [−.0056, .0093]	−.0035 [−.0168, .0091]	−.0010 [−.0085, .0062]
Qwen3	−.0054 [−.0157, .0049]	−.0091 [−.0197, .0015]	−.0073 [−.0151, .0006]
GLM-4.7-Flash	−.0005 [−.0109, .0100]	.0022 [−.0084, .0130]	.0009 [−.0062, .0080]
Lightning	.0017 [−.0026, .0061]	−.0018 [−.0072, .0037]	−.0001 [−.0035, .0034]
Nano	.0018 [−.0012, .0047]	−.0008 [−.0038, .0022]	.0005 [−.0017, .0026]
Gemma4	−.0106 [−.0328, .0111]	−.0016 [−.0202, .0180]	−.0061 [−.0208, .0087]
OLMoE	.0061 [−.0011, .0132]	.0039 [−.0049, .0127]	.0050 [−.0011, .0110]
GPT-OSS-20B	−.0004 [−.0049, .0042]	−.0029 [−.0103, .0043]	−.0017 [−.0060, .0027]
GPT-OSS-120B	.0031 [−.0093, .0156]	.0092 [−.0044, .0228]	.0061 [−.0025, .0150]
MoE output			
Qwen3.6	.0009 [−.0063, .0080]	−.0115 [−.0240, .0003]	−.0053 [−.0120, .0010]
Qwen3	.0498 [.0321, .0676]	.0540 [.0305, .0771]	.0519 [.0339, .0698]
GLM-4.7-Flash	.0802 [.0598, .1006]	.0482 [.0316, .0647]	.0642 [.0475, .0806]
Lightning	.0006 [−.0039, .0052]	−.0012 [−.0064, .0040]	−.0003 [−.0041, .0036]
Nano	.0018 [−.0015, .0051]	−.0010 [−.0048, .0029]	.0004 [−.0024, .0032]
Gemma4	.1430 [.0727, .2295]	.0946 [.0459, .1458]	.1188 [.0681, .1793]
OLMoE	.0402 [.0250, .0556]	.0296 [.0123, .0472]	.0349 [.0211, .0487]
GPT-OSS-20B	.0031 [−.0094, .0154]	−.0059 [−.0167, .0048]	−.0014 [−.0108, .0079]
GPT-OSS-120B	.0433 [.0290, .0581]	.0439 [.0268, .0617]	.0436 [.0312, .0564]
Residual			
Qwen3.6	−.0096 [−.0308, .0130]	.0356 [.0101, .0625]	.0130 [−.0054, .0327]
Qwen3	.7282 [.6313, .8269]	.5921 [.4989, .6879]	.6601 [.5712, .7509]
GLM-4.7-Flash	.1989 [.1594, .2404]	.1371 [.1077, .1673]	.1680 [.1353, .2019]
Lightning	.1492 [.1285, .1704]	.1125 [.0953, .1304]	.1309 [.1127, .1495]
Nano	.0793 [.0680, .0907]	.0624 [.0525, .0726]	.0709 [.0608, .0811]
Gemma4	1.1705 [.8775, 1.4791]	.2586 [.1150, .4163]	.7145 [.5321, .9069]
OLMoE	.1368 [.0859, .1925]	.1586 [.1049, .2169]	.1477 [.0966, .2038]
GPT-OSS-20B	.1352 [.1016, .1696]	.1019 [.0726, .1323]	.1186 [.0909, .1473]
GPT-OSS-120B	.2262 [.1895, .2645]	.1654 [.1342, .1977]	.1958 [.1659, .2271]

Table 5: Controls for full-step all-router interventions and directly paired contrasts. All panels use $N = 290$, $K = 235$ and 50,000 source-component resamples. Norm-random interventions match the paired donor displacement separately at each token position and layer. OLMoE's (†) paired-minus-norm-random interval lies close to zero; a larger matched-donor T does not imply more correctness-related information.

Model	Paired T	Paired−matched	Paired−norm-random
Qwen3.6	.347 [.286, .408]	−.168 [−.227, −.111]	.218 [.170, .267]
Qwen3	1.576 [1.358, 1.796]	−.497 [−.663, −.335]	.651 [.505, .804]
GLM-4.7-Flash	.231 [.189, .275]	−.071 [−.102, −.040]	.165 [.131, .200]
Lightning	.221 [.186, .256]	−.073 [−.096, −.050]	.175 [.144, .208]
Nano	.143 [.118, .170]	−.040 [−.056, −.024]	.120 [.098, .143]
Gemma4	1.208 [.958, 1.459]	−.926 [−1.199, −.655]	.915 [.689, 1.141]
OLMoE†	.1852 [.1200, .2527]	−.0168 [−.0648, .0304]	.0466 [.0003, .0938]
GPT-OSS-20B	.297 [.242, .353]	−.161 [−.214, −.109]	.189 [.146, .233]
GPT-OSS-120B	.428 [.350, .509]	−.169 [−.229, −.111]	.214 [.155, .275]

$[-0.03077, 0.01299]$, whereas $D_{\mathrm{E}} = 0.03560$ $[0.01006, 0.06248]$. Their direct difference is -0.04524 $[-0.07478, -0.01557]$, establishing receiver dependence. Its D_{common} has median -0.00077 and 10%-trimmed mean 0.00217, showing that the positive mean does not correspond to a uniform pairwise direction.

The remaining eight residual panels have positive intervals for both receiver types, although their magnitudes need not match. For example, the correct-minus-error receiver differences for Qwen3 and GLM are 0.1362 $[0.0690, 0.2045]$ and 0.0618 $[0.0376, 0.0874]$, respectively.

Coverage and position. Expanding token coverage from the last token to the full step does not produce a common trend in the residual-minus-all-router gap. The gap increases in Qwen3.6, Qwen3, Gemma 4, and GPT-OSS 20B; decreases in GLM, Lightning, Nano, and GPT-OSS 120B; and is unresolved in OLMoE.

Position is more consistent. With token count fixed at eight, moving from suffix-8 to divergence-8 reduces the residual-minus-all-router gap in all nine checkpoints. This pattern does not imply a larger absolute routing response near divergence. In GPT-OSS 120B, the all-router divergence-minus-suffix difference is -0.0028 $[-0.0591, 0.0535]$, whereas the residual difference is -0.1946 $[-0.2315, -0.1589]$. The changed ordering there is driven primarily by residual sensitivity to token position.

Qwen3.6's full-step residual effect also differs from its heavy-tailed last-token result: its mean is 1.2821, median 1.2086, 10%-trimmed mean 1.2246, and 88.3% of pair effects are positive. These statistics summarize the same distribution of pair-level effects; the pair-weighted mean remains the primary summary.

Control comparisons. Full-step all-router paired-minus-matched intervals are negative in eight checkpoints and include zero in OLMoE. By contrast, paired-minus-norm-random intervals are positive in all nine checkpoints, although OLMoE lies close to zero: 0.0466 $[0.0003, 0.0938]$, with norm-random $T = 0.1387$ and paired-donor $T = 0.1852$.

For the all-layer MoE-output intervention, every paired-minus-norm-random interval is positive, whereas every paired-minus-matched interval includes zero. The latter result does not establish equivalence between paired and matched external donors.

Two GPT-OSS checkpoints. Matched-pair 120B-minus-20B contrasts are 0.1316 $[0.0704, 0.1935]$ for full-step all-router T and -0.8498 $[-0.9704, -0.7294]$ for full-step residual T. Single-layer D_{common} differences are 0.0450 $[0.0290, 0.0618]$ at the MoE-output interface and 0.0773 $[0.0414, 0.1138]$ at the residual interface. Effects therefore do not increase uniformly across interfaces in the larger checkpoint.

Table 6: All-layer MoE-output interventions at the last token. Paired-donor T is positive and exceeds the norm-matched random control in all nine models. "Paired" uses the corresponding correct/error trace as the donor, whereas "matched" uses an external donor from another problem matched by error type and boundary position. All paired-minus-matched intervals include zero. Entries are pair means with pointwise 95% source-component percentile intervals ($N = 290$, $K = 235$, 50,000 resamples).

Model	Paired-donor T	Paired−matched	Paired−norm-random
Qwen3.6	.099 [.064, .138]	.019 [−.007, .048]	.088 [.051, .130]
Qwen3	.933 [.785, 1.083]	.084 [−.022, .190]	.848 [.713, .984]
GLM-4.7-Flash	.162 [.127, .198]	.014 [−.012, .040]	.149 [.116, .183]
Lightning	.214 [.181, .249]	.019 [−.010, .048]	.196 [.164, .229]
Nano	.101 [.083, .120]	.008 [−.006, .022]	.088 [.071, .106]
Gemma4	1.133 [.869, 1.422]	.136 [−.107, .388]	.938 [.684, 1.217]
OLMoE	.113 [.067, .162]	.007 [−.035, .050]	.093 [.055, .135]
GPT-OSS-20B	.388 [.331, .446]	.030 [−.015, .077]	.364 [.280, .447]
GPT-OSS-120B	.563 [.478, .650]	.022 [−.045, .091]	.528 [.446, .612]

The two checkpoints also differ in local anchor, routed depth, expert count, training, and checkpoint identity. These comparisons therefore do not isolate parameter count or define a scaling law.

E Verification Performance, Decision Flips, and Normalized Effects

All three quantities reported here are computed from the saved candidate scores of the nine-checkpoint window runs and, where stated, of the full-step all-router fixed-receiver run; no additional model forwards are required. A receiver receives the verdict `INCORRECT` when $M > 0$ and `CORRECT` when $M \leq 0$, and the truth is `CORRECT` for the correct receiver and `INCORRECT` for the error receiver. These are within-study values of a fixed candidate-scoring verdict on the paired PRMBench assay used in this study, not official PRMBench benchmark scores. The mean-logprob margin of Equation 1 is primary, and the total-logprob margin gives the same picture, with accuracy within 0.021 and AUROC within 0.002 of the primary values. Exact ties at zero margin occur only on the total metric, in 5 baseline receivers of Qwen3, and are read as `CORRECT`. Intervals follow the rest of the paper: two-sided pointwise 95% source-component percentile intervals from 50,000 resamples without multiplicity correction, with the AUROC interval computed from resample-weighted tie-aware rank comparisons.

E.1 Unpatched verification performance

Accuracy is the fraction of the 580 receivers whose verdict matches the truth. Error recall is the fraction of error receivers called `INCORRECT`, and the correct-receiver false-positive rate is the fraction of correct receivers called `INCORRECT`. AUROC ranks receivers by M with the error receiver as the positive class. The mean gap $g = M(x_{\mathrm{E}}) - M(x_{\mathrm{C}})$ averages the within-pair margin difference. Labels are balanced, so chance is 0.5 for accuracy and for AUROC, and balanced accuracy equals accuracy.

All nine checkpoints are above chance (Table 7). Accuracy ranges from 0.5672 in OLMoE to 0.7862 in Gemma and AUROC from 0.6018 to 0.8744. Two checkpoints are strongly biased in opposite directions: Nano calls 0.621 of correct receivers `INCORRECT`, whereas OLMoE never does and recalls 0.134 of errors. Within the window runs, unpatched reruns reproduce these scores exactly, with zero verdict flips in every checkpoint.

E.2 Decision flip rates

For each receiver we compare the unpatched verdict with the patched verdict. The four transition classes are correct-to-correct, correct-to-wrong, wrong-to-correct, and wrong-to-wrong, where correct

Table 7: Unpatched verification performance on the paired PRMBench assay used in this study, under the mean-logprob margin and the $M > 0$ verdict rule. Accuracy, error recall, and the correct-receiver false-positive rate use all 580 receivers of the 290 pairs; AUROC takes the error receiver as the positive class; g is the within-pair baseline margin gap. These are within-study assay values, not official PRMBench benchmark scores. Intervals are two-sided pointwise 95% source-component percentile intervals from 50,000 resamples ($N = 290$, $K = 235$).

Model	Accuracy	Error recall	Correct-receiver FPR	AUROC	Mean gap g
Qwen3.6	0.7776	0.6517	0.0966	0.8580	1.5549
	[0.7457, 0.8090]	[0.5951, 0.7075]	[0.0605, 0.1359]	[0.8279, 0.8866]	[1.3789, 1.7296]
Qwen3	0.7345	0.5586	0.0897	0.8610	4.6818
	[0.7038, 0.7650]	[0.5000, 0.6151]	[0.0559, 0.1263]	[0.8316, 0.8891]	[4.1758, 5.1846]
GLM-4.7-Flash	0.5845	0.1862	0.0172	0.8058	0.8147
	[0.5621, 0.6075]	[0.1424, 0.2319]	[0.0035, 0.0339]	[0.7748, 0.8360]	[0.7125, 0.9175]
Lightning	0.7621	0.7517	0.2276	0.8475	0.8680
	[0.7298, 0.7937]	[0.7013, 0.8007]	[0.1771, 0.2800]	[0.8185, 0.8755]	[0.7716, 0.9655]
Nano	0.6534	0.9276	0.6207	0.8171	0.5024
	[0.6241, 0.6838]	[0.8953, 0.9570]	[0.5586, 0.6812]	[0.7857, 0.8478]	[0.4378, 0.5689]
Gemma4	0.7862	0.8103	0.2379	0.8744	7.0863
	[0.7552, 0.8165]	[0.7651, 0.8537]	[0.1852, 0.2941]	[0.8470, 0.9003]	[6.3870, 7.7836]
OLMoE	0.5672	0.1345	0.0000	0.6018	0.6056
	[0.5481, 0.5875]	[0.0962, 0.1751]	[0.0000, 0.0000]	[0.5701, 0.6334]	[0.4427, 0.7738]
GPT-OSS-20B	0.7155	0.8034	0.3724	0.8202	1.2942
	[0.6858, 0.7451]	[0.7562, 0.8486]	[0.3132, 0.4339]	[0.7901, 0.8497]	[1.1356, 1.4525]
GPT-OSS-120B	0.7103	0.8138	0.3931	0.7785	1.3498
	[0.6781, 0.7425]	[0.7662, 0.8596]	[0.3345, 0.4539]	[0.7460, 0.8104]	[1.1818, 1.5193]

and wrong refer to agreement with the truth. The flip rate is the fraction of receivers whose verdict changes. The harmful rate conditions on baseline-correct receivers and the beneficial rate on baseline-wrong receivers; the two have different denominators and are never subtracted. The change in accuracy equals the unconditional beneficial rate minus the unconditional harmful rate. By construction, the paired design supplies the opposite-role donor, so a donor-aligned flip is a flip away from the truth, and accuracy decreases in every paired residual panel and in every full-step all-router panel. Where flips are few the sign is not forced: in 41 of the remaining 198 paired panels, all of them router-logit, MoE-output, or non-full-step all-router panels, accuracy instead increases by at most 0.0091, a net gain of five receivers or fewer. Positive mean T indicates an average margin shift toward the opposite-role donors' labels, but does not determine the accuracy change.

Under the paired donor at full step, the all-router intervention changes 0.0621 (OLMoE) to 0.2293 (GPT-OSS-120B) of verdicts, the norm-random control 0.0310 to 0.1121, and the matched external donor 0.0690 to 0.3517, exceeding the paired donor in all nine checkpoints (Table 8). The paired residual intervention changes 0.0672 to 0.5362 of verdicts and separates into two regimes: Qwen3.6, Qwen3, Gemma, and GPT-OSS-20B fall to patched accuracy 0.2724 to 0.3741, well below chance, whereas GLM-4.7-Flash and OLMoE end at about chance (0.4966 and 0.5000) and Lightning, Nano, and GPT-OSS-120B at 0.5517 to 0.6931; these five lose 0.0672 to 0.1017 of accuracy. Single-layer router-logit interventions flip 0.0052 to 0.0259 of verdicts at full step. Conditional beneficial rates are undefined for OLMoE on correct receivers, whose false-positive rate is exactly zero, and rest on one to five receivers for GLM-4.7-Flash; the unconditional rates and accuracy changes are defined throughout.

The same reading applies to the full-step all-router fixed-receiver run of Appendix F. Holding the receiver fixed and switching the donor from correct to error changes 0.0172 (Nano) to 0.1517 (Qwen3.6) of verdicts, far fewer than patching itself. On correct receivers, GPT-OSS-20B moves 0.1000 of verdicts to `INCORRECT` and 0.0414 to `CORRECT`, and Gemma 0.0897 and 0.0586, the categorical counterpart of their positive correct-receiver D. By contrast, the accuracy change on error

Table 8: Full-step verdict flips under the mean-logprob margin, on all 580 receivers of the 290 pairs. The first three columns give the all-router flip rate under the paired donor, a matched external donor, and the norm-random control; the last two give the paired residual flip rate and its accuracy change. Self-patch controls produce zero flips in every checkpoint. Intervals are two-sided pointwise 95% source-component percentile intervals from 50,000 resamples ($N = 290$, $K = 235$).

Model	All-router paired	All-router matched	All-router norm-random	Residual paired	Residual Δaccuracy
Qwen3.6	0.1690	0.2690	0.0879	0.5362	-0.5052
	$[0.1399, 0.1993]$	$[0.2352, 0.3032]$	$[0.0651, 0.1123]$	$[0.4826, 0.5890]$	$[-0.5629, -0.4467]$
Qwen3	0.2155	0.3517	0.1121	0.4397	-0.4155
	$[0.1831, 0.2491]$	$[0.3134, 0.3903]$	$[0.0850, 0.1406]$	$[0.3867, 0.4929]$	$[-0.4711, -0.3596]$
GLM-4.7-Flash	0.0966	0.1155	0.0586	0.0983	-0.0879
	$[0.0722, 0.1224]$	$[0.0896, 0.1421]$	$[0.0402, 0.0782]$	$[0.0732, 0.1242]$	$[-0.1150, -0.0618]$
Lightning	0.2034	0.3000	0.0586	0.1103	-0.0690
	$[0.1701, 0.2377]$	$[0.2607, 0.3402]$	$[0.0412, 0.0769]$	$[0.0847, 0.1381]$	$[-0.0956, -0.0423]$
Nano	0.1414	0.1897	0.0466	0.1121	-0.1017
	$[0.1107, 0.1738]$	$[0.1544, 0.2266]$	$[0.0295, 0.0654]$	$[0.0857, 0.1398]$	$[-0.1301, -0.0742]$
Gemma4	0.1879	0.3224	0.0552	0.4362	-0.4121
	$[0.1567, 0.2202]$	$[0.2819, 0.3635]$	$[0.0372, 0.0748]$	$[0.3866, 0.4859]$	$[-0.4653, -0.3582]$
OLMoE	0.0621	0.0690	0.0310	0.0672	-0.0672
	$[0.0437, 0.0816]$	$[0.0498, 0.0893]$	$[0.0179, 0.0451]$	$[0.0448, 0.0914]$	$[-0.0914, -0.0448]$
GPT-OSS-20B	0.1621	0.3172	0.0466	0.4224	-0.4155
	$[0.1313, 0.1941]$	$[0.2752, 0.3595]$	$[0.0308, 0.0636]$	$[0.3681, 0.4775]$	$[-0.4710, -0.3603]$
GPT-OSS-120B	0.2293	0.3086	0.1052	0.0983	-0.0845
	$[0.1926, 0.2668]$	$[0.2683, 0.3493]$	$[0.0804, 0.1311]$	$[0.0741, 0.1237]$	$[-0.1096, -0.0603]$

receivers has the same sign and a similar size under the correct and the error donor in every checkpoint, for example 0.0552 versus 0.0690 in Lightning and -0.1724 versus -0.1517 in Qwen3.6. Baseline-to-patched flips in that run are therefore dominated by patching itself rather than by donor role.

E.3 Effects normalized by the baseline gap

We normalize each panel by the baseline margin gap of the same eligible pairs under the same margin, $R = \mathrm{mean}(T)/\mathrm{mean}(g)$ with $g = M(x_{\mathrm{E}}) - M(x_{\mathrm{C}})$. By Equation 5, $2T = g - g_{\mathrm{after}}$ with $g_{\mathrm{after}} = M(x_{\mathrm{E}\leftarrow\mathrm{C}}) - M(x_{\mathrm{C}\leftarrow\mathrm{E}})$, so $R = 0.5$ means that the mean post-transfer gap is zero and $R = 1$ an equal-magnitude reversal. R is a ratio of means. It can be negative or exceed one, it is not a mediation share or an explained variance, and it differs from dividing each receiver's effect by its own distance to the decision boundary, which is not computed here.

Full-step residual R runs from 0.2371 in GPT-OSS-120B to 0.9040 in GPT-OSS-20B (Table 9). Its interval lies entirely above 0.5 in GPT-OSS-20B, Qwen3, Qwen3.6, and Gemma, whose mean post-transfer gaps are -1.0455, -3.7285, -1.0094, and -2.6533. GLM-4.7-Flash (0.5177 $[0.4857, 0.5496]$) and OLMoE (0.5356 $[0.4726, 0.6084]$) sit at about 0.5, and Lightning, Nano, and GPT-OSS-120B near 0.25. Full-step all-router R runs from 0.1705 to 0.3365, reducing the mean gap to between 33% and 66% of baseline, and single-layer router-logit R stays within $[-0.0094, 0.0167]$ in all windows.

Within a checkpoint and margin definition, normalization preserves interface orderings when the panels share an eligible-pair set and a positive mean baseline gap. Across windows or controls with different eligible-pair sets, the denominator is recomputed and orderings need not be preserved. On the residual interface the matched external donor gives essentially the same R as the paired donor, for example 0.2512 versus 0.2503 in Lightning, 0.2589 versus 0.2586 in Nano, and 0.2482 versus 0.2371 in GPT-OSS-120B, so normalization does not change the donor-generic character of that effect. All 1,152 panels have denominators whose intervals exclude zero. OLMoE has non-positive

Table 9: Paired-donor effects normalized by the baseline margin gap of the same pairs, $R = \mathrm{mean}(T)/\mathrm{mean}(g)$, under the mean-logprob margin. The last column gives the fraction of pairs with a non-positive baseline gap. Intervals are two-sided pointwise 95% source-component percentile intervals from 50,000 resamples ($N = 290$, $K = 235$).

Model	Mean gap g	R, all-router (full step)	R, residual (full step)	R, residual (last token)	Pairs with $g \leq 0$
Qwen3.6	1.5549	0.2229	0.8246	0.0291	0.1379
	[1.3789, 1.7296]	[0.1933, 0.2523]	[0.7991, 0.8505]	[0.0088, 0.0514]	
Qwen3	4.6818	0.3365	0.8982	0.0892	0.0897
	[4.1758, 5.1846]	[0.3057, 0.3666]	[0.8786, 0.9170]	[0.0737, 0.1050]	
GLM-4.7-Flash	0.8147	0.2839	0.5177	0.2856	0.1586
	[0.7125, 0.9175]	[0.2481, 0.3179]	[0.4857, 0.5496]	[0.2566, 0.3149]	
Lightning	0.8680	0.2541	0.2503	0.1726	0.1345
	[0.7716, 0.9655]	[0.2257, 0.2821]	[0.2351, 0.2657]	[0.1602, 0.1854]	
Nano	0.5024	0.2856	0.2586	0.1726	0.1655
	[0.4378, 0.5689]	[0.2534, 0.3168]	[0.2412, 0.2764]	[0.1612, 0.1845]	
Gemma4	7.0863	0.1705	0.6872	0.1171	0.0931
	[6.3870, 7.7836]	[0.1388, 0.2025]	[0.6520, 0.7226]	[0.0891, 0.1478]	
OLMoE	0.6056	0.3059	0.5356	0.1577	0.3966
	[0.4427, 0.7738]	[0.2277, 0.3846]	[0.4726, 0.6084]	[0.1005, 0.2200]	
GPT-OSS-20B	1.2942	0.2293	0.9040	0.1134	0.1310
	[1.1356, 1.4525]	[0.1932, 0.2663]	[0.8819, 0.9268]	[0.0927, 0.1357]	
GPT-OSS-120B	1.3498	0.3173	0.2371	0.1550	0.1931
	[1.1818, 1.5193]	[0.2747, 0.3589]	[0.2156, 0.2602]	[0.1386, 0.1724]	

baseline gaps in 0.3966 of pairs, against at most 0.1931 elsewhere, so its R should not be read as a per-pair recovery fraction.

E.4 Unpatched scores of the full-step all-router run

The unpatched candidate scores of the run in Appendix F reproduce the window-run scores receiver by receiver for Qwen3.6, Qwen3, OLMoE, GPT-OSS-20B, and GPT-OSS-120B, but not for GLM-4.7-Flash, Lightning, Nano, and Gemma. Recorded inputs, tokenization, pair identities, and receiver roles agree across the two runs for all nine checkpoints, but the source of the score discrepancies remains unresolved. Unpatched accuracy differs by at most 0.006. Table 7 reports the window-run scores; each run retains its own baseline. The fixed-receiver D estimates of Appendix F compare patched scores within the wide-D run and do not use the window-run baseline.

F Full-Step All-Router Fixed-Receiver Analysis

F.1 Experimental design and primary analysis

This experiment extends the fixed-receiver contrast in Equation 6 to every adapter-defined routed layer listed in Table 2 and every token in the evaluated step, with $\alpha = 1$. All nine checkpoints use the same 290 receiver pairs, 235 source components, and a fixed assignment to 204 distinct external donor pairs. Matching excludes the receiver pair and its source component, then prioritizes agreement in error type and proximity of the first-error position.

For each receiver, the correct and error members of the matched external donor pair separately replace router logits across all routed layers and the full step. Donor and receiver spans are aligned using the monotone mapping defined in Appendix C. Each checkpoint retains its native BF16 or MXFP4 execution and native expert dispatch.

The primary endpoint remains the candidate-wise mean-logprob verification margin in Equation 1. For each receiver pair, the four donor/receiver combinations yield D_{C}, D_{E}, and their within-pair average D_{common}. All four intervention scores and identity controls are generated under the same loaded model invocation. All 290 pairs are complete for every checkpoint; no values are imputed.

A positive D_{common} indicates an average correctness-dependent effect across the two receiver types. It does not require both receiver-specific effects, or every individual pair effect, to be positive.

The primary uncertainty analysis uses 50,000 draws from `numpy.random.Generator(PCG64)` with seed 20260908. Each draw samples 235 receiver source components with replacement, retains all pairs belonging to each sampled component, and divides the summed pair effects by the sampled pair count. Both receiver roles share each draw, so D_{common} has $N = 290$, not 580. Donors are neither rematched nor resampled independently.

All intervals are two-sided pointwise 95% percentile intervals conditional on the discovery panel and fixed donor assignment, without simultaneous coverage or multiplicity adjustment. Table 10 reports the primary results; Table 11 reports the scoring sensitivity. Counts of intervals excluding zero should not be interpreted as estimates of a model-population success rate.

Table 10: Primary full-step all-router fixed-receiver effects. Each checkpoint uses $N = 290$ receiver pairs, $K = 235$ source components, and the same fixed assignment to 204 external donor pairs. Entries are mean-logprob estimates with two-sided pointwise 95% source-component percentile intervals from 50,000 PCG64 resamples at $\alpha = 1$. D_{common} averages the two receiver types within pair. Checkpoint-specific margin scales should not be interpreted as ability rankings.

Model	D_{correct}	D_{error}	D_{common}
Qwen3.6	0.0198	0.0012	0.0105
	[−0.0199, 0.0595]	[−0.0616, 0.0644]	[−0.0282, 0.0486]
Qwen3	−0.0954	−0.0738	−0.0846
	[−0.2095, 0.0192]	[−0.2102, 0.0589]	[−0.1872, 0.0166]
GLM-4.7-Flash	0.0041	−0.0292	−0.0125
	[−0.0273, 0.0369]	[−0.0572, −0.0011]	[−0.0363, 0.0110]
Lightning	0.0213	0.0043	0.0128
	[−0.0005, 0.0430]	[−0.0197, 0.0280]	[−0.0050, 0.0304]
Nano	0.0032	−0.0094	−0.0031
	[−0.0109, 0.0171]	[−0.0219, 0.0028]	[−0.0134, 0.0069]
Gemma4	0.3875	0.0540	0.2207
	[0.0762, 0.7002]	[−0.2159, 0.3335]	[−0.0114, 0.4545]
OLMoE	−0.0088	0.0415	0.0164
	[−0.0609, 0.0435]	[−0.0435, 0.1270]	[−0.0376, 0.0713]
GPT-OSS-20B	0.1137	0.0262	0.0699
	[0.0480, 0.1803]	[−0.0335, 0.0871]	[0.0181, 0.1226]
GPT-OSS-120B	−0.0032	−0.0071	−0.0051
	[−0.0489, 0.0435]	[−0.0539, 0.0396]	[−0.0393, 0.0290]

F.2 Candidate weighting sensitivity

Total log probability uses the same candidate scores as the primary endpoint but weights candidates by their token counts. It is therefore a scoring sensitivity rather than an independent replication.

Gemma illustrates the distinction. All 1,160 four-cell scoring rows contain two target tokens for `CORRECT` and three for `INCORRECT`. Let A denote the error-minus-correct donor change in the `INCORRECT` candidate's mean log probability, averaged first over the two receivers and then over pairs, and let B denote the corresponding change for `CORRECT`. Then

$$D_{\mathrm{common}}^{\mathrm{mean}} = A - B, \qquad D_{\mathrm{common}}^{\mathrm{total}} = 3A - 2B. \tag{9}$$

Here $A = 0.10743$ and $B = -0.11327$. The primary mean-logprob result is 0.2207 $[-0.0114, 0.4545]$, whereas the total-logprob result is 0.5488 $[0.0034, 1.1020]$. The secondary analysis verifies this reweighting for every pair and every bootstrap draw, with maximum absolute residual 1.78×10^{-15}. The change is not a uniform positive rescaling, which would preserve whether an interval contains zero.

Gemma's correct-receiver interval is positive under both endpoints, while its primary D_{common} interval includes zero. GLM's error-receiver interval is slightly negative under mean scoring and includes zero under total scoring. Mean scoring remains the primary endpoint for all checkpoints.

Table 11: Total-logprob sensitivity for the same full-step all-router scores, pairs, donor assignments, and bootstrap draws as Table 10. Candidate totals change the relative weighting of verdict strings when their token counts differ. These results do not replace the primary mean-logprob endpoint and are not an independent replication.

Model	D_{correct}	D_{error}	D_{common}
Qwen3.6	0.0477	-0.0055	0.0211
	$[-0.0541, 0.1509]$	$[-0.1502, 0.1411]$	$[-0.0710, 0.1118]$
Qwen3	-0.2638	-0.1974	-0.2306
	$[-0.5815, 0.0554]$	$[-0.5702, 0.1644]$	$[-0.5154, 0.0491]$
GLM-4.7-Flash	0.0147	-0.0774	-0.0313
	$[-0.0753, 0.1090]$	$[-0.1552, 0.0009]$	$[-0.0986, 0.0356]$
Lightning	0.0717	0.0040	0.0379
	$[-0.0052, 0.1485]$	$[-0.0775, 0.0838]$	$[-0.0239, 0.0990]$
Nano	0.0090	-0.0307	-0.0108
	$[-0.0375, 0.0549]$	$[-0.0702, 0.0082]$	$[-0.0442, 0.0217]$
Gemma4	0.9582	0.1395	0.5488
	$[0.2172, 1.7054]$	$[-0.4944, 0.7933]$	$[0.0034, 1.1020]$
OLMoE	-0.0254	0.1043	0.0394
	$[-0.1765, 0.1265]$	$[-0.1255, 0.3361]$	$[-0.1092, 0.1911]$
GPT-OSS-20B	0.2588	0.0719	0.1653
	$[0.1017, 0.4193]$	$[-0.0600, 0.2074]$	$[0.0463, 0.2868]$
GPT-OSS-120B	0.0136	-0.0245	-0.0055
	$[-0.0968, 0.1269]$	$[-0.1340, 0.0832]$	$[-0.0859, 0.0749]$

F.3 Post hoc direct contrasts and common-input identity

After inspecting the nine-checkpoint results, we computed two classes of secondary direct contrast from the saved four-cell scores: $D_{\text{C}} - D_{\text{E}}$ within each checkpoint and GPT-OSS 20B minus 120B in D_{common}. Each difference is formed within receiver pair before the same 50,000-component resampling, jointly resampling both receiver types or both checkpoints.

Table 12 reports these contrasts. GPT-OSS 20B has $D_{\text{C}} - D_{\text{E}} = 0.0874$ $[0.0158, 0.1597]$. Gemma and GLM have direct receiver-difference intervals that include zero; different zero-crossing behavior of their marginal intervals therefore does not establish a receiver-type difference. Conversely, an interval containing zero does not establish equivalence.

All nine checkpoints share receiver sample identities, source components, and fixed donor assignments. For the two GPT-OSS checkpoints, the secondary analysis additionally compares 1,160 receiver/donor identity occurrences and 2,320 candidate-conditioned complete inputs. It checks saved rendered-prompt hashes, input formats and hashes, step spans, sample/source identities, target IDs and positions, tensor shapes, and hashes recomputed from the complete `input_ids` arrays.

The full input-identity projection has the same SHA-256 for both checkpoints:

`35c1fb039b9cfa0617565d039650f3d8f6`
`b00e4016a835ec4f40bc703416a136`.

The archived identity records do not separately store rendered prompt text, but equality of the saved whole-prompt hashes and freshly hashed complete token arrays establishes identity of the full inputs used in this comparison. This check applies only to the full-step fixed-receiver GPT comparison and does not establish identity with earlier T, local D, or representation-probe experiments.

The paired GPT-OSS 20B-minus-120B contrast is 0.0751 $[0.0206, 0.1301]$. It compares two fixed checkpoints under common inputs and donor assignments. Training, checkpoint identity, expert count, and routed-layer coverage still differ, so the result does not isolate parameter count.

Similarly, comparing an unresolved local-router interval with a positive full-step all-router interval is not a direct test of the effect of expanding coverage. Full-step router and local residual interventions also differ in layer and token coverage. Neither comparison identifies an equal-coverage interface difference, a directional fraction D/T, or a compression component $T - D$. The all-layer MoE-output intervention has no corresponding fixed-receiver D experiment.

Table 12: Post hoc mean-logprob contrasts for full-step all-router D. Receiver-type differences are correct minus error within checkpoint; the checkpoint difference is GPT-OSS 20B minus 120B in D_{common}. Differences are formed within pair and jointly resampled over the same 235 receiver source components ($N = 290$, fixed donors; 50,000 PCG64 draws). Intervals are two-sided pointwise 95% percentile intervals without multiplicity adjustment. The GPT checkpoint contrast uses verified common inputs but does not isolate parameter count.

Model / checkpoint contrast	Estimate	95% CI	Bootstrap SE
Receiver-role difference: $D_{\text{correct}} - D_{\text{error}}$			
Qwen3.6	0.0185	$[-0.0545, 0.0897]$	0.0369
Qwen3	-0.0216	$[-0.1624, 0.1248]$	0.0733
GLM-4.7-Flash	0.0332	$[-0.0033, 0.0712]$	0.0190
Lightning	0.0170	$[-0.0115, 0.0461]$	0.0147
Nano	0.0126	$[-0.0041, 0.0295]$	0.0086
Gemma4	0.3335	$[-0.0215, 0.6877]$	0.1804
OLMoE	-0.0503	$[-0.1387, 0.0406]$	0.0459
GPT-OSS-20B	0.0874	$[0.0158, 0.1597]$	0.0368
GPT-OSS-120B	0.0039	$[-0.0589, 0.0673]$	0.0324
Checkpoint difference in D_{common}			
GPT-OSS-20B − GPT-OSS-120B	0.0751	$[0.0206, 0.1301]$	0.0279

F.4 Execution checks

All nine checkpoints completed preflight, smoke, and full execution stages with accepted atomic-pair, identity-control, and native-dispatch checks. Gemma's resource interruption retained 44 complete pairs from invocation `29687b42-ec5c-4c96-9129-02e06d0136a2`; the resumed invocation `4e46dce0-861b-44c9-8ace-f0f1d925818b` supplied the remaining 246. An unfinished pair was recomputed in full rather than joining intervention cells across invocations.

Execution version, scientific source, and configuration remained unchanged. The recovery procedure does not alter the receiver-pair estimand or the source-component resampling unit. Recorded full-run and smoke forwards have a lower bound of 63,488. Work interrupted before completion is not recorded exactly, so the fresh-plan count of 63,216 should not be interpreted as an actual execution total.

The secondary verification reconstructed all 54 original mean/total endpoints and reproduced their stored point estimates, interval endpoints, and bootstrap standard errors before computing additional

contrasts. The manuscript analysis reuses this accepted primary analysis and introduces no additional model forwards or protected-validation access. Artifact identities are listed in Appendix J.

G Representation-Probe Analysis

Error-related information and intervention effects remain distinct. We first ask whether routing states contain information that distinguishes correct from erroneous steps, and whether such information predicts the effect of intervening on routing. Using out-of-fold L2-logistic probes grouped by source problem, router-logit representations distinguish the two classes across eight evaluated checkpoints. In the historical six-model panel, router-logit AUROC ranges from 0.801–0.884, compared with 0.829–0.907 for residual states and 0.837–0.915 for router inputs. The GLM and GPT-OSS-120B extensions show the same pattern, with router-logit AUROCs of 0.818 and 0.860, respectively (details below).

A readable signal does not by itself determine intervention strength. GLM, for example, has a router-logit AUROC of 0.818 but an unresolved local router D and a positive full-step all-router T. GPT-OSS-120B similarly contains a strong correct/error signal in its router logits while its broad routing intervention behaves differently from its local residual intervention. Thus, information that can be read from a routing representation does not by itself determine how changing that representation will affect the verification judgment. Protocol-specific probe details and cross-checkpoint comparability limitations are reported below.

G.1 Probe protocols and checkpoint details

The representation-probe analysis asks whether internal activations contain information that distinguishes correct from erroneous steps. The probe uses L2-regularized logistic classifiers with training-fold standardization and out-of-fold predictions. Folds group the 290 paired examples by canonical source-problem component so that linked examples are not split across training and evaluation folds.

The six-checkpoint probe panel uses a post hoc mapping of the 290 pairs to 235 source components and is therefore retrospective.

The positive class is the error-bearing step. AUROC measures how well a representation separates correct from erroneous steps; it does not measure causal use of that information or identify a correctness-specific representation. We probe residual states, router inputs, router logits, and expert-aligned routing representations.

Table 13: Middle-anchor matched out-of-fold AUROC under 235 canonical source-problem groups. "Access/causal layer" gives the observational recapture anchor and the headline intervention anchor, respectively. "Expert weight" is the expert-ID-aligned sparse gate-weight representation. These are access measurements at capture interfaces, not intervention effects or a mediation sequence.

Model	Protocol grade	Access / causal layer	Residual	Router input	Router logits	Expert weight
Gemma 4	modern	15 / 15	0.893	0.901	0.868	0.792
Nemotron-3 Nano	modern	27 / 27	0.855	0.865	0.831	0.688
Nemotron-3.5 Lightning	modern	27 / 27	0.865	0.862	0.826	0.733
Qwen3.6-35B-A3B	modern	20 / 20	0.871	0.873	0.815	0.790
GPT-OSS 20B	extended	12 / 12	0.829	0.837	0.801	0.715
Qwen3-30B-A3B	historical retrospective	24 / 36	0.907	0.915	0.884	0.750
OLMoE-1B-7B	unavailable	— / 8	—	—	—	—

Across the six checkpoints in the shared probe table, residual and router-input representations generally have higher AUROC than router logits, which in turn exceed expert-aligned sparse-weight representations. A narrower 238-group recapture-native construction changes every cell by at most 0.031 and preserves this ordering.

Qwen3 uses a different probe/intervention layer pair (L24/L36) and is therefore not treated as an equal-layer intervention comparison. OLMoE has no corresponding recapture manifest or activation tensors and is omitted rather than assigned an imputed value. No additional contrast-feature control is constructed because the archived manifests do not contain predeclared topic, language, token-class, or formatting labels.

GLM. The GLM L24 source-group out-of-fold analysis uses 290 pairs, 580 rows, and 235 source components. Residual AUROC is 0.8646 $[0.8387, 0.8918]$, router-input AUROC is 0.8717 $[0.8451, 0.8973]$, router-logit AUROC is 0.8180 $[0.7864, 0.8476]$, expert-aligned sparse-weight AUROC is 0.7427 $[0.7082, 0.7775]$, and expert-membership AUROC is 0.7010 $[0.6644, 0.7397]$.

The direct router-input-minus-router-logit difference is 0.0537 $[0.0293, 0.0781]$; residual-minus-input is unresolved at -0.0071 $[-0.0161, 0.0017]$. Intervals condition on fitted out-of-fold predictions and use 1,000 source-component bootstrap draws with seed 20260829. These probe intervals are separate from the 50,000-draw intervention analyses. GLM's separate 64-pair behavior subset has AUROC 0.8401.

GPT-OSS 120B. The GPT-OSS 120B probe uses the saved L18 recaptures. Residual AUROC is 0.8813 $[0.8543, 0.9059]$, router-input AUROC is 0.8847 $[0.8580, 0.9075]$, router-logit AUROC is 0.8599 $[0.8310, 0.8868]$, expert-aligned sparse-weight AUROC is 0.7039 $[0.6678, 0.7433]$, and expert-membership AUROC is 0.6704 $[0.6326, 0.7128]$.

The direct input-minus-logit difference is 0.0248 $[0.0046, 0.0446]$, logit-minus-aligned-weight is 0.1560 $[0.1132, 0.1920]$, and aligned-weight-minus-membership is 0.0335 $[0.0207, 0.0474]$. Residual-minus-input is unresolved at -0.0034 $[-0.0112, 0.0042]$.

The analysis reuses 290 pairs, 580 rows, 235 source components, and five grouped folds, with training-fold preprocessing and L2-logistic fitting. Intervals condition on the fitted out-of-fold predictions and use 1,000 source-component draws with seed 20260829.

The GPT-OSS 20B and 120B recaptures share raw prompt content, pair identities, and verification boundaries but differ in the rendered Harmony system-header date. Their rendered-prompt and token-input hashes therefore differ for all 580 rows. Together with checkpoint and architecture differences, this prevents interpreting their probe difference as a causal effect of model scale.

Text baseline on the same folds. A fixed word unigram/bigram TF–IDF baseline uses the boundary-step text and the same L2-logistic settings as the router probes ($C = 1$, balanced classes, `liblinear`, 2,000 maximum iterations), without tuning. Vocabulary and IDF are fit only within each training fold. The comparison uses all 580 samples, 290 pairs, 235 source components, and the same five folds. Identical text inputs in the five paired-first-error protocols yield one fitted baseline reused across checkpoints.

Table 14: Boundary-step text baseline on the identical 580 samples, 235 source components, and five saved interface-access folds. Differences compare the middle router-logit probe with the fixed text baseline; 95% intervals use paired source-component resampling.

Model (access layer)	Text AUROC [95% CI]	Router minus text [95% CI]
Gemma 4 (L15)	0.743 $[0.710, 0.777]$	$+0.124$ $[0.084, 0.159]$
Nemotron-3 Nano (L27)	0.743 $[0.710, 0.777]$	$+0.088$ $[0.051, 0.123]$
Nemotron-3.5 Lightning (L27)	0.743 $[0.710, 0.777]$	$+0.083$ $[0.043, 0.116]$
Qwen3.6-35B-A3B (L20)	0.743 $[0.710, 0.777]$	$+0.072$ $[0.031, 0.110]$
GPT-OSS-20B (L12)	0.743 $[0.710, 0.777]$	$+0.057$ $[0.024, 0.092]$
Historical boundary sensitivity		
Qwen3-30B-A3B (L24)	0.734 $[0.696, 0.771]$	$+0.150$ $[0.110, 0.190]$

Table 14 reports router-logit comparisons with this text baseline. Intervals jointly resample 235 source components 1,000 times with seed 20260829 while conditioning on the fitted out-of-fold predictions; the classifiers are not refit within each draw. Qwen3's distinct boundary construction remains separate.

A router advantage over this TF–IDF baseline does not establish a correctness-specific representation or remove all surface confounds. Contextual router states contain information unavailable to isolated boundary-step word features, and the representation dimensions and classifier capacities are not matched.

In Qwen3's separate matched surface comparison, routing and text features reach AUROCs of 0.777 and 0.795, with paired difference -0.019 $[-0.047, 0.010]$. This result is compatible with readable routing information but does not establish causal or selectively used routing information.

The representation-probe and intervention analyses measure different objects. The probes passively read residual, router-input, router-logit, and expert-aligned routing representations, whereas the intervention analyses replace activations at adapter-defined boundaries and measure changes in the verification margin. Similar interface names do not make these measurements stages of a serial mediation pathway.

H Natural-Error Evaluation on ProcessBench

The interface difference also appears on naturally occurring errors. We next test whether the local residual-versus-router difference extends beyond the constructed PRMBench pairs. For this exploratory extension, we construct 64 cross-problem pairs from ProcessBench (Zheng et al., 2025), each matching a trace with a naturally occurring first error to an all-correct trace from the same generator at the same step index. We evaluate GPT-OSS-20B, Gemma, and Nemotron-3 Nano using the same local residual and router-logit interventions.

Residual interventions produce larger absolute changes than router-logit interventions in all six model-by-direction comparisons. The paired residual-minus-router T difference is also positive in all three models: 0.0476 $[0.0145, 0.0817]$ for GPT-OSS-20B, 0.1241 $[0.0246, 0.2402]$ for Gemma, and 0.0575 $[0.0396, 0.0755]$ for Nano. Five of the six residual point estimates move in the direction associated with the replacement activation's correctness; Gemma's correct-to-error direction is the exception. An exploratory analysis relates this exception to whether the receiver initially missed the error, but this analysis is observational and subject to baseline conditioning effects (Section H.2).

These results provide complementary evidence that the local interface difference is not limited to the constructed PRMBench errors. However, ProcessBench uses only three models and 64 cross-problem pairs and does not include the fixed-receiver D or token-window analyses. We therefore treat it as a separate natural-error extension rather than part of the primary nine-model comparison.

H.1 Evaluation setup and supplementary results

The ProcessBench extension tests whether the local residual-versus-router difference also appears for naturally occurring reasoning errors. From the public GSM8K subset, 193 of 400 source traces are all-correct and 207 contain a first-error label. We construct 64 same-generator pairs, each linking a trace with a naturally occurring first error to an all-correct trace at the same step index. The resulting 128 traces contain no trace reuse, and all 64 pairs are cross-problem matches.

GPT-OSS-20B, Gemma, and Nemotron-3 Nano use the same processed-data selection, binary verification prompt, boundary policy, two replacement directions, dose $\alpha = 1$, identity conditions, and outcome metric. The reported tables use a 50,000-resample analysis of the immutable pair-level scores. Architecture-specific anchors remain unchanged. A scoped Nemotron adapter repair changed neither the samples, boundaries, estimand, nor intervention comparisons.

On the balanced 128-trace endpoint, GPT-OSS has accuracy 0.625 and AUROC 0.736; Gemma has accuracy 0.688 and AUROC 0.854; and Nemotron-3 Nano has accuracy 0.641 and AUROC 0.824. Each full run contains 128 behavior rows, 128 recapture rows, and 768 intervention rows. Artifact-integrity and coverage checks are independent of effect direction and statistical significance.

Residual directional intervals exclude zero in four of the six checkpoint-by-direction comparisons, whereas router intervals exclude zero in none. Five of six residual point estimates move in the direction associated with the correctness of the replacement activation. Gemma's correct-to-error direction is the exception. Nano has the expected sign in both directions; GPT-OSS is clear only

Table 15: ProcessBench natural-error transfer on 64 same-generator, cross-problem pairs per model. MC→FE uses a matched-correct donor and first-error receiver; FE→MC swaps their roles. Each cell shows an estimate above its paired-cluster percentile-bootstrap 95% interval (50,000 resamples; no multiplicity adjustment). T and G come directly from paired rows, not marginal intervals.

Model	Site	MC→FE	FE→MC	T	$G = T_{\text{res}} - T_{\text{router}}$
GPT-OSS 20B	residual	$-.0590$ $[-.1029, -.0163]$	$+.0242$ $[-.0344, .0824]$	.0416 $[.0065, .0775]$	.0476 $[.0145, .0817]$
	router	$+.0051$ $[-.0060, .0164]$	$-.0070$ $[-.0174, .0035]$	$-.0060$ $[-.0143, .0025]$	
Gemma 4 26B-A4B	residual	$+.0442$ $[-.0813, .1594]$	$+.2948$ $[.1424, .5035]$	.1253 $[.0224, .2452]$	.1241 $[.0246, .2402]$
	router	$+.0010$ $[-.0339, .0358]$	$+.0034$ $[-.0203, .0268]$	.0012 $[-.0183, .0205]$	
Nemotron-3 Nano	residual	$-.0517$ $[-.0686, -.0345]$	$+.0667$ $[.0454, .0897]$	.0592 $[.0410, .0778]$	.0575 $[.0396, .0755]$
	router	$-.0033$ $[-.0093, .0027]$	$+.0001$ $[-.0054, .0056]$	.0017 $[-.0024, .0056]$	

in the correct-to-error direction; Gemma has a large expected effect in the reverse direction. These differences motivate the receiver-state analysis below.

H.2 Receiver-state analysis

We reanalyze the same 64 pairs in all three checkpoints while retaining both replacement directions and both intervention sites. Receivers are grouped by whether their unpatched verification margin correctly judges the evaluated boundary. Group intervals condition on the observed group sizes: 50,000 PCG64 draws with seed 0 independently resample pairs within the two groups, and group differences are computed from the joint draws.

These exploratory, unadjusted intervals describe an observational grouping. The baseline margin used to define the groups also appears with a negative sign in the intervention effect, so the resulting group comparisons are subject to baseline conditioning.

Figure 5 shows that Gemma's positive correct-to-error point estimate is not attributable to a single pair: 37/64 effects are positive, the median is 0.0781, and all 64 leave-one-out means lie between 0.0281 and 0.0752.

The mean is 0.3320 for the 11 initially missed errors and -0.0155 for the 53 initially detected errors, a difference of 0.3475 $[0.1342, 0.5708]$. This does not imply that every positive effect belongs to the missed-error group; 28/53 detected-error receivers also have positive effects. Table 16 reports the complete two-direction decomposition.

Table 16: Exploratory residual-transfer effects by the receiver's unpatched judgment. Entries give mean shifts in nats/target-token; the group contrast is misjudged minus correctly judged, with marginal 95% intervals from 50,000 within-group pair-bootstrap draws. C→E is matched correct to first error; E→C is the reverse. All original pairs remain included.

Model	Direction	Correctly judged n, mean	Misjudged n, mean	Group contrast [95% CI]
GPT-OSS	C→E	46, $-.0776$	18, $-.0114$	.0662 [$-.0249$, .1564]
	E→C	34, .0436	30, .0022	$-.0415$ [$-.1618$, .0747]
Gemma 4	C→E	53, $-.0155$	11, .3320	.3475 [.1342, .5708]
	E→C	35, .3126	29, .2733	$-.0392$ [$-.4320$, .2568]
Nano	C→E	59, $-.0574$	5, .0164	.0738 [.0272, .1340]
	E→C	23, .1018	41, .0469	$-.0549$ [$-.1085$, $-.0053$]

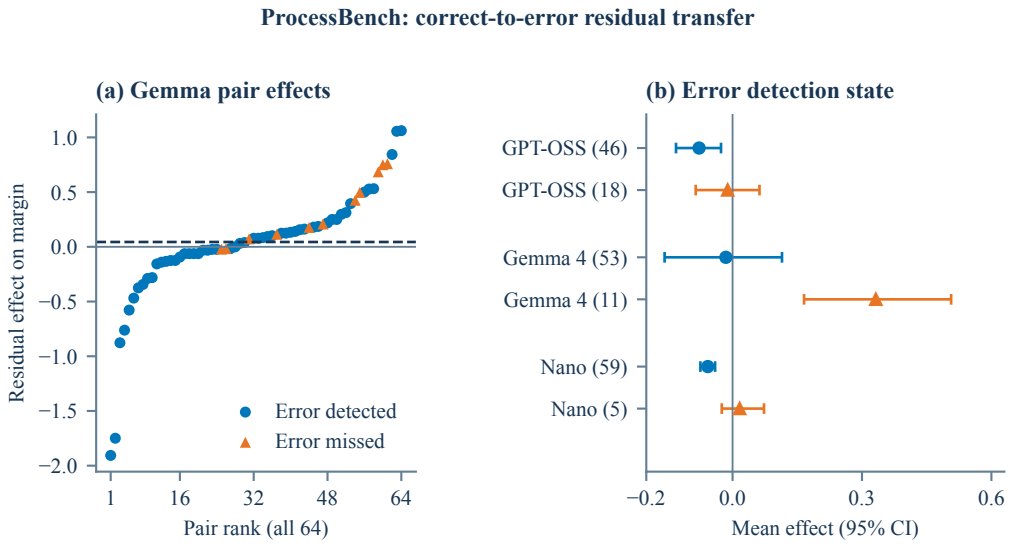

Figure 5: Exploratory correct-to-error residual intervention on ProcessBench. (a) All 64 Gemma pair effects, ordered by value; the dashed line is their mean. (b) Mean effects grouped by whether the receiver initially detects the error, with sample counts in parentheses. Intervals use 50,000 percentile bootstrap draws within the observed receiver-state groups. Negative effects correspond to the expected direction for this intervention.

Construction and shared-problem sensitivity. The 64 pairs cover 12 generators and first-error step indices zero through five, with counts 8, 17, 18, 12, 5, and 4. Although 49 pairs have equal full-trace step counts, the median absolute difference in visible step-text length is 176 characters; matching the step index therefore does not match the amount of visible text.

The 128 traces represent 125 normalized problems, linking the pairs into 61 connected components: 59 singletons, one two-pair component, and one three-pair component. A sensitivity analysis jointly resamples these components, retains every member pair, and computes pair-weighted means.

With 50,000 PCG64 draws and seed 0, residual-minus-router gaps are 0.0476 $[0.0126, 0.0831]$ for GPT-OSS, 0.1241 $[0.0239, 0.2412]$ for Gemma, and 0.0575 $[0.0389, 0.0759]$ for Nano. These supplementary intervals preserve the positive-gap result without replacing the canonical pair-bootstrap analysis. Gemma's receiver-state contrast is 0.3475 $[0.1299, 0.5779]$ under the same component resampling.

Routing changes and signed averages. Gemma's near-zero mean router effect averages over pair-level changes with opposing signs. In the correct-to-error direction, 30 pair effects are positive, 33 are negative, and one is zero. Their mean absolute magnitude is 0.1108, compared with a signed mean of 0.0010; the residual mean absolute magnitude is 0.3197.

A small signed router average therefore does not imply that every individual router intervention leaves the verification judgment unchanged. Native top-k changes and output magnitudes remain descriptive manipulation diagnostics rather than calibrated causal doses or mediation quantities. No additional donor controls or error-type labels are introduced in this analysis.

I Additional Controls and Implementation Checks

I.1 Identity controls

Unpatched reruns and self-patch conditions verify that the intervention machinery preserves the receiver when no cross-example activation replacement is made. Failure of an identity condition would invalidate the corresponding cross-example intervention rows. Passing the identity

check is an implementation validation and is not evidence that the natural causal contribution of the intervention site is zero.

I.2 Native routing execution

The intervention adapters retain each checkpoint's native expert selection, normalization, and dispatch after router-logit replacement. Execution artifacts record patched logits together with the downstream expert IDs and weights used for dispatch. Machine checks compare these quantities under each checkpoint's native top-k and normalization rules. These checks are manipulation validation rather than cross-checkpoint effect-size covariates.

ProcessBench separately contains 256/256 verified dispatch rows per checkpoint. Cross-example router interventions change native top-k selection in 49/64 GPT-OSS, 63/64 Gemma, and 61/64 Nano rows in each direction.

I.3 Matched-donor construction and interpretation

Matched external donors are selected by prioritizing agreement in error type and proximity of the first-error position, with a seeded hash used only for tie-breaking. The matched pair's correct and error members supply the same donor roles as the corresponding paired-donor condition. Donor assignments are fixed for analysis, and matched donors come from different source-problem components.

These controls test whether using the paired donor produces a larger response than using a matched external donor. The primary comparison is the direct paired-minus-matched difference. A relative comparison, when reported in the archived artifacts, divides the difference of means by the paired-donor effect; it is not a mean of pairwise ratios and should not be interpreted as the fraction of an underlying correctness mechanism.

Positive T can still arise from compression of the original correct/error judgment gap (Equation 5). Matched-donor controls do not remove this ambiguity because donor and receiver roles still change together. The fixed-receiver experiments address the separate question of whether the intervention effect changes when the receiver is held constant and the replacement activation changes from a correct to an erroneous step. Neither analysis decomposes T into mechanistic components.

I.4 Norm-random controls

The norm-random intervention adds a Gaussian direction to the receiver activation, scaled to the ℓ_2 norm of the paired donor–receiver displacement at the corresponding intervention site. It tests whether an intervention of similar magnitude but random direction produces a comparable response. A smaller norm-random effect does not exclude structured but correctness-independent activation replacement.

For full-step all-router interventions, paired-minus-norm-random intervals are positive in all nine checkpoints, with OLMoE close to zero. For the all-layer MoE-output intervention, every paired-minus-norm-random interval is positive. By contrast, paired-minus-matched-donor intervals for the all-layer MoE-output intervention all include zero. These controls therefore show effects beyond random displacement magnitude without resolving paired-example specificity.

J Reproducibility and Artifact Records

The nine-checkpoint inputs come from three campaign archives: `campaign_20260907_A` (Qwen3.6, Qwen3, GLM), `campaign_20260907_B` (Lightning, Nano, Gemma, OLMoE), and `campaign_20260907_C` (GPT-OSS 20B and 120B). Their resolved configurations, executed adapters, scores, donor manifests, token geometry, and native analyses define the experimental identities.

The `abc_synthesis_20260908` CPU analysis reads these artifacts, reads the saved model outputs and computes the common-window and direct-contrast analyses. It executes no new model forwards and does not alter the discovery selection. The protected validation split is not used.

The versioned window JSON has SHA-256 `75431e13cb3cf0ebba9f8a86df443fcdd a5c7fd10b6cd2fb28fd390909d241ac`; the D/bundle headline CSV has SHA-256

0be3afb97f570165d0d7a0e662ba32c2f
b9c6f8b5e58f55222ce13c86c636971. Machine-readable rows include the source file, analysis version, and JSON pointer or pair identity.

Full-step all-router D records. The separate campaign_20260908_D/final_v1 archive retains each checkpoint's atomic four-cell scores, full input identities, resolved configuration, adapter and runner hashes, native 50,000-draw analysis, and recovery records.

Its wide_d_nine_model_v1.json summary has SHA-256

6b1ccd80ecd58d01025ee41f3b2c7be2
74875a8315090182440450l9fa98e14b.

The wide_d_synthesis_20260908 analysis stores the main summary and a separately identified wide_d_posthoc_secondary_contrasts_v1 artifact. Its receipt records source-member hashes, the complete GPT input-identity projection, Gemma candidate-weight checks, and exact reproduction of the original endpoints. These records do not overwrite the A/B/C campaign artifacts or the representation-probe inputs. No protected-validation split was used in the full-step D experiment or its secondary analysis.

Representation-probe and ProcessBench analyses have separate input and analysis records. Exact model revisions, pair manifests, fold assignments, seeds, and source hashes are stored in the analysis archives.

K Claim Boundaries

The experiments support four main conclusions within the tested settings. First, post-MoE residual interventions produce larger local responses than router-logit and MoE-output interventions across all nine checkpoints and all four token windows. Second, expanding routing interventions across routed layers and tokens can substantially increase their effect and can change the ordering relative to a local residual intervention. Third, fixed-receiver experiments show correctness-dependent local residual effects in eight checkpoints, whereas the corresponding broad-routing effects vary across checkpoints and receiver types. Fourth, all-layer MoE-output interventions exceed norm-matched random controls, while the current data do not resolve an advantage of paired donors over matched external donors.

More specifically, the full-step all-router fixed-receiver analysis provides a positive D_{common} interval for GPT-OSS 20B and a positive correct-receiver interval for Gemma. Post hoc direct contrasts support a larger correct-receiver effect for GPT-OSS 20B and a larger D_{common} for GPT-OSS 20B than GPT-OSS 120B. Qwen3.6 remains an explicit exception to the common local-residual fixed-receiver pattern.

These conclusions retain the intervention interface, layer and token coverage, receiver type, verification endpoint, and checkpoint scope under which they were measured. Intervals containing zero do not establish equivalence. The representation-probe and ProcessBench analyses provide complementary evidence under their own protocols and are not pooled with the nine-checkpoint intervention study.

The experiments do not establish that routing is behaviorally irrelevant, that any intervention has exactly zero effect, that the tested interfaces form a formal mediation pathway, that residual states are the unique mechanism of verification, that the observed effects isolate a pure correctness representation, that paired-example identity is required for an intervention effect, or that the findings hold universally across MoE models and verification tasks.