FROM REPRESENTATION CORRESPONDENCE TO COMPUTATIONAL TRAJECTORIES IN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Geometric analyses of neural networks characterize correspondences between hidden states, but it is unclear when a correspondence between representations extends to a correspondence between the computations that transform them. We study this question by distinguishing a state correspondence at one depth from the transport of a trajectory by a depth-indexed family of correspondences.

Across equivalent number representations, affine low-rank correspondences accurately transport hidden states but fail when applied statically to local residual-stream updates. Transporting the two endpoints of an update with the correspondence fitted at each endpoint instead raises update alignment from 0.096 to 0.683, approaching an update-specific correspondence at 0.692. The same matched-sample pattern replicates across musical-pitch notation systems ($0.083 \rightarrow 0.762$, versus 0.759 for an update-specific map). In numbers, decomposing the dynamic predictor identifies evolution of the state correspondence across adjacent layers as the dominant additional component. Independent adjacent-map estimation reduces the effect but retains 47.1% of the static-to-update-specific gap, while shuffled correspondences collapse to near zero.

We then ask whether depth-evolving state correspondences predict computation outside the examples used to fit them. In controlled addition, correspondences learned only from baseline states of training operands predict arithmetic-specific trajectories for held-out operands. A strong late-layer correspondence emerges between Question and Compute formulations, and transported Question–Compute states induce aligned arithmetic-specific updates when executed by the actual destination transformer block.

Finally, a semantic-shuffle control separates native computational correspondence from local dynamical compatibility: shuffling identity pairings destroys state alignment, native trajectory prediction, and interventional update recovery, while substantial layerwise square closure remains. Thus a representational snapshot does not generally specify a computational trajectory, but the evolution of representation correspondences across depth can contain predictive information about that trajectory. Computational correspondence should therefore be validated against native computation rather than inferred from state alignment or local dynamical compatibility alone.

1 INTRODUCTION

Geometric structure in hidden states has become a central language for interpreting neural networks. Directions, subspaces, manifolds, and alignment maps describe how concepts are organized inside a model. These objects answer an important question: *how are internal states related?* A stronger question is whether a correspondence between states also identifies a correspondence between the computations that transform them.

Consider matched residual-stream states in two domains A and B. At layer ℓ, let an affine correspondence be $T_\ell(x) = L_\ell x + c_\ell$, and let $V_A^\ell = h_A^{\ell+1} - h_A^\ell$ denote the local update. A natural static prediction is that $L_\ell V_A^\ell$ should align with the corresponding update in B. We find that this prediction can fail even when T_ℓ accurately aligns the states. The key point is that the correspondence itself

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.

may evolve with depth. The displacement between mapped endpoints is

$$T_{\ell+1}(h_A^{\ell+1}) - T_\ell(h_A^\ell), \tag{1}$$

which need not equal $L_\ell V_A^\ell$. A single representational snapshot and a depth-indexed family of correspondences therefore make different predictions about computation.

This motivates our central question: *when does a correspondence between representations extend to a correspondence between computations?* We distinguish four increasingly stronger properties: alignment of matched states, transport of their trajectories across depth, the update induced when a transported state is executed by the destination network, and local dynamical compatibility of the transported paths. Our experiments show that these properties are related but not interchangeable.

We first study equivalent numerical representations—digits, number words, and ordinals—where cross-format state correspondence is strong. A state map fitted at a single layer transports local updates poorly, whereas transporting the two endpoints with their respective layerwise maps nearly reaches an update-specific transport control under matched-sample estimation. The same static-to-dynamic recovery appears in an independent musical-pitch domain spanning scientific, lexical, and Helmholtz notation. In numbers, an exact decomposition attributes most of the recovered alignment to evolution of the state correspondence across adjacent layers. Cross-fitting adjacent maps on disjoint halves reduces the effect, but still recovers 47.1% of the finite-sample-matched static-to-update-specific gap and remains far above a shuffled-pairing control.

We next ask whether this depth-evolving correspondence predicts computation outside the examples used to estimate it. In controlled addition, correspondences fitted only from baseline states of training operands predict arithmetic-specific trajectories for unseen operands. A strong late-layer correspondence emerges between Question and Compute formulations, whereas pairs involving Instruction remain substantially weaker. Moreover, when transported Question–Compute states are inserted into the destination residual stream, the actual destination transformer block produces arithmetic-specific updates aligned with native destination updates.

Finally, we separate native computational correspondence from local dynamical compatibility. A semantically shuffled correspondence can retain substantial agreement between the two paths of the corresponding layerwise dynamical square even while state alignment, native trajectory prediction, and interventional update recovery collapse. Local dynamical consistency therefore does not by itself identify the intended computation.

Contributions. We make three main contributions.

1. We formulate computational correspondence as a depth-dependent transport problem and show that evolving state correspondences recover update transport missed by a frozen single-layer map. The matched-sample recovery replicates across number and musical-pitch domains, and in numbers a cross-fitted audit retains 47.1% of the static-to-update-specific gap.
2. We demonstrate out-of-sample and interventional computational correspondence in arithmetic: maps learned only from baseline states predict arithmetic-specific trajectories for unseen operands, and transported states induce aligned updates under the actual destination transformer block.
3. We distinguish native computational correspondence from local dynamical compatibility. Semantically shuffled correspondences can retain substantial layerwise square closure while losing state alignment, native trajectory prediction, and interventional update recovery.

2 Related Work

Representation geometry and transformations. Language models exhibit structured geometry for numerical and other semantic variables. Numerical information is decodable and organized across surface forms (Levy & Geva, 2025; Kadlčík et al., 2025; Štefánik et al., 2026), while broader work describes concepts using directions, subspaces, manifolds, and context-dependent transformations (Gurnee & Tegmark, 2024; Shafran et al., 2026; Modell, 2025; Hu et al., 2026). These results

establish activation geometry as a meaningful description of internal organization. Our question concerns what is required to extend such a description across depth. Rather than asking only whether matched states admit a correspondence T_ℓ, we ask whether the layer-indexed family $\{T_\ell\}$ transports the trajectories generated by the network. This distinguishes correspondence at a representational snapshot from correspondence of the computation connecting successive representations.

Computational correspondence and causal abstraction. Causal-abstraction approaches distinguish representational alignment from stronger claims about computation by testing whether aligned internal variables exhibit corresponding causal behavior under intervention (Geiger et al., 2021; 2022; 2024). Related work on distributed alignment search further allows such correspondences to occupy distributed subspaces rather than individual neurons (Geiger et al., 2024). Our setting is complementary. We do not begin with a specified high-level causal model whose variables are to be aligned with a neural network. Instead, we compare two realizations within the same network and ask whether an empirically learned state correspondence extends to their native layer-to-layer computations. This leads naturally to a depth-dependent compatibility relation,

$$F_B^\ell \circ T_\ell \approx T_{\ell+1} \circ F_A^\ell, \tag{2}$$

whose form is analogous to an intertwining or equivariance condition. Unlike architectural equivariance, however, the correspondence is learned post hoc rather than imposed by a known symmetry. Moreover, our shuffled controls show that approximate local compatibility of this form need not identify the intended native computational correspondence.

Arithmetic representations, mechanisms, and trajectories. Arithmetic provides a controlled setting in which representation and computation can be studied together. Prior work has decoded operands, intermediate quantities, and results from hidden states and localized arithmetic behavior to components, circuits, or sparse features (Hanna et al., 2023; Quirke & Barez, 2024; Yan, 2025; Sun et al., 2025; Polyakov et al., 2025; Du et al., 2025). Recent work further examines whether arithmetic mechanisms persist across symbolic, natural-language, and other surface forms (Polyakov et al., 2025; Naganna et al., 2026). Our experiments ask a different question: whether correspondences learned between baseline representations predict the layer-to-layer trajectories induced by the same arithmetic operation on unseen operands. Thus component overlap, behavioral equivalence, state correspondence, and trajectory correspondence are treated as separate empirical properties.

Layer-specific readouts provide another precedent for treating hidden-state coordinates as depth dependent. The Tuned Lens (Belrose et al., 2023) fits a separate affine decoder at each transformer block to track how latent predictions evolve across depth. Our setting differs in both the objects being aligned and the question being asked: rather than mapping each layer into a common output space, we learn correspondences between two surface-form representations at each depth and ask whether the *change* in those correspondences transports the native update between adjacent layers.

3 METHODS

3.1 STATE CORRESPONDENCES AND STATIC TRANSPORT

For surface-form domain A, concept identity c, and residual-stream layer ℓ, define

$$X_A^\ell(c) = h_A^\ell(c), \qquad V_A^\ell(c) = h_A^{\ell+1}(c) - h_A^\ell(c). \tag{3}$$

All geometric objects are fitted on training identities and evaluated on held-out identities. Hidden states are extracted either at the final target token or by mean pooling over the target span.

At each layer, we fit an affine low-rank correspondence between matched states,

$$T_\ell^{A\to B}(x) = (x - \mu_{X,A}^\ell)P_{X,\ell}^{A\to B} + \mu_{X,B}^\ell, \qquad P_{X,\ell}^{A\to B} = B_{X,A}^\ell R_{X,\ell}^{A\to B}(B_{X,B}^\ell)^\top, \tag{4}$$

where $B_{X,A}^\ell$ and $B_{X,B}^\ell$ are PCA bases of the centered training clouds and $R_{X,\ell}^{A\to B}$ is an orthogonal Procrustes map fitted between their coordinates. We fit an analogous update-specific correspondence

$$P_{V,\ell}^{A\to B} = B_{V,A}^\ell R_{V,\ell}^{A\to B}(B_{V,B}^\ell)^\top \tag{5}$$

as a positive control.

The simplest trajectory prediction treats the state correspondence as fixed over the update:

$$\widehat{V}^{\ell}_{B,\mathrm{static}} = V^{\ell}_{A} P^{A\to B}_{X,\ell}. \tag{6}$$

Predictions and target updates are centered using training identities and evaluated by mean rowwise cosine similarity on held-out identities. Additional static diagnostics—including representation-support energy, same-support update refits, and matched-support representation/update maps—are defined in Appendix B.2.

3.2 Depth-evolving trajectory transport

A layer-indexed state correspondence predicts the displacement between mapped endpoints,

$$\widehat{V}^{\ell}_{B,\mathrm{dyn}} = T^{A\to B}_{\ell+1}(X^{\ell+1}_{A}) - T^{A\to B}_{\ell}(X^{\ell}_{A}). \tag{7}$$

For affine maps $T_\ell(X) = L_\ell X + c_\ell$, this predictor decomposes exactly as

$$\begin{aligned}\widehat{V}^{\ell}_{B,\mathrm{dyn}} = {} & L_\ell V^{\ell}_{A} + (L_{\ell+1} - L_\ell) X^{\ell}_{A} \\ & + (L_{\ell+1} - L_\ell) V^{\ell}_{A} + (c_{\ell+1} - c_\ell).\end{aligned} \tag{8}$$

The first term is the static prediction; the remaining terms arise from the evolution of the state correspondence across the two endpoints.

Importantly, accurate endpoint transport alone does not establish computational correspondence: differencing two successful endpoint predictions can mechanically approximate the intervening update. We therefore test whether depth-evolving correspondences predict trajectories not used to fit the maps, using independent adjacent-map estimation below and baseline-only correspondences in the arithmetic experiments.

We compare static and depth-evolving predictions with an update-specific positive control fitted directly between V^{ℓ}_{A} and V^{ℓ}_{B}. Unless otherwise stated, all reported transport scores are centered rowwise cosines on held-out identities.

3.3 Independent adjacent-map estimation

Because T_ℓ and $T_{\ell+1}$ fitted from the same identities can share estimation error, we independently estimate adjacent maps using random half-splits of the training identities. In the SAME condition, both maps are fitted on the same half. In the CROSSFIT condition, T_ℓ and $T_{\ell+1}$ are fitted on disjoint halves; we evaluate both half assignments and average them symmetrically. A SHUFFLE control breaks the source–destination identity pairing when fitting the adjacent map while preserving the marginal state clouds.

Cross-fitting reduces the number of examples available to each map. We therefore fit the update-specific positive control using the same half-sample budget as each representation map. This provides a finite-sample-matched reference for the fraction of update transport recovered by the depth-evolving predictor. The full original training set is used only as a train-side centering pool and does not contribute target updates to map fitting.

3.4 Experimental settings

The controlled number experiments use values 0–99 represented as Arabic digits, English number words, and ordinals, with a fixed $70/30$ identity split shared across domains. Headline number analyses use Llama-3.1-8B, mean pooling, rank-32 correspondences, and middle-depth layers 7–24. We independently repeat the matched-sample static and depth-evolving transport analysis on musical pitch, using 60 pitch identities expressed in scientific, lexical, and Helmholtz notation. The number domain is used for the independent adjacent-map cross-fit audit and the decomposition of map evolution. Static-transport robustness across model families, pooling choices, and ranks is reported in the appendix.

Active-computation experiments use controlled addition under Question, Compute, and Instruction formulations as described below. Exact prompt templates, musical-pitch construction, layer conventions, and uncertainty procedures are provided in Appendix B.

3.5 ARITHMETIC TRAJECTORY TRANSPORT AND INTERVENTION

Let $H_p^\ell(n, \Delta)$ denote the residual-stream state for operand n, offset Δ, and prompt formulation p, and define

$$V_p^\ell(n, \Delta) = H_p^{\ell+1}(n, \Delta) - H_p^\ell(n, \Delta). \tag{9}$$

We isolate the arithmetic-specific update by subtracting the matched $+0$ trajectory,

$$dV_p^\ell(n, \Delta) = V_p^\ell(n, \Delta) - V_p^\ell(n, 0). \tag{10}$$

For each directed formulation pair $A \to B$, the baseline-only correspondence $T_\ell^{A\to B}$ is fitted exclusively from $+0$ states of training operands. Evaluation uses nonzero additions on held-out operands, so neither the evaluated operand states nor their arithmetic-specific updates are used to fit the correspondence.

This makes trajectory transport an out-of-sample computational prediction: the maps are fitted only from baseline states of training operands, whereas evaluation uses arithmetic-specific updates of held-out operands whose nonzero-addition trajectories never contribute to correspondence fitting.

We use rank 128 for the arithmetic correspondence, compared with rank 32 in the controlled number analysis, because the arithmetic maps align contextualized states across substantially different prompt formulations rather than alternative surface forms of the same isolated identity. We use the larger rank as the primary arithmetic setting and treat rank as an analysis setting rather than as a quantity directly comparable across the two experiments.

The depth-evolving prediction is baseline-subtracted in the same way:

$$\begin{aligned} d\widehat{V}_{B,\mathrm{dyn}}^\ell(n, \Delta) = &\Big[T_{\ell+1}(H_A^{\ell+1}(n, \Delta)) - T_\ell(H_A^\ell(n, \Delta))\Big] \\ &- \Big[T_{\ell+1}(H_A^{\ell+1}(n, 0)) - T_\ell(H_A^\ell(n, 0))\Big]. \end{aligned} \tag{11}$$

We evaluate its centered rowwise cosine with the native destination update $dV_B^\ell(n, \Delta)$.

We next test the correspondence interventionally. At destination block ℓ, we replace only the final prompt-token residual state by the transported source state

$$H_B^\ell(n, \Delta) \longleftarrow T_\ell^{A\to B}(H_A^\ell(n, \Delta)), \tag{12}$$

while leaving all other sequence positions and the destination prompt context unchanged. We then execute the actual destination transformer block and compare the resulting arithmetic-specific update with the native destination update. The matched $+0$ intervention is subtracted to isolate the arithmetic-specific effect.

Finally, we measure local dynamical compatibility through the approximate layer-dependent relation

$$F_B^\ell \circ T_\ell \approx T_{\ell+1} \circ F_A^\ell, \tag{13}$$

evaluated at the arithmetic-specific update level. To distinguish such local compatibility from native computational correspondence, we fit a semantic shuffle control in which destination operand identities are permuted by the same fixed permutation at every layer. This preserves marginal state distributions, train/test splits, map rank, and depth consistency while destroying the intended source–destination identity correspondence. Additional endpoint and trajectory diagnostics and uncertainty procedures are reported in Appendix C.

4 RESULTS

4.1 STATIC MAPS FAIL, DEPTH-EVOLVING CORRESPONDENCES RECOVER UPDATE TRANSPORT

A strong state correspondence does not imply that the same map transports the local update. In the number domain, held-out cross-format state transport reaches 0.954, but applying the representation-derived map statically to updates reaches only

$$C_{\mathrm{static}} = 0.0963.$$

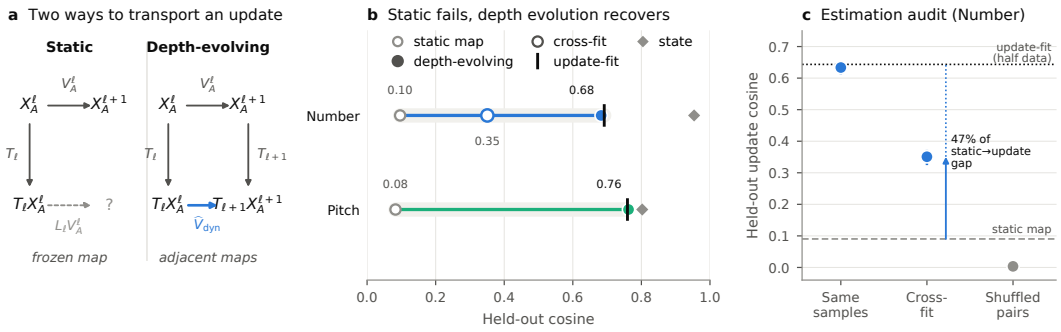

Figure 1: **A frozen state correspondence misses update transport, while its evolution across depth recovers it.** **(a)** A static prediction applies the linear part of a single map, $L_\ell V_A^\ell$, whereas depth-evolving transport maps the two endpoints with adjacent correspondences, $T_{\ell+1}(X_A^{\ell+1}) - T_\ell(X_A^\ell)$. **(b)** On held-out identities, the static map transports updates poorly in both number and musical-pitch domains, while depth-evolving transport approaches the update-specific positive control. The number cross-fit point uses independently fitted adjacent maps; pitch is shown under matched-sample fitting. State correspondence is shown for scale. **(c)** Number-domain finite-sample audit. Dynamic alignment drops from same-half to cross-fitted adjacent maps but remains far above shuffled pairings. Relative to the matched half-sample static and update-specific references, cross-fitting recovers 47.1% of the available gap. Error bars are the empirical 2.5–97.5 percentile range across 50 random splits, not independent-sample confidence intervals.

Transporting the two endpoints with their respective layerwise correspondences changes the result substantially:

$$C_{\text{dynamic}} = 0.6826, \qquad C_{\text{update}} = 0.6917,$$

where C_{update} is an update-specific positive control. Under matched-sample estimation this corresponds to

$$R_{\text{recovered}} = \frac{C_{\text{dynamic}} - C_{\text{static}}}{C_{\text{update}} - C_{\text{static}}} = 0.985.$$

The result is consistent across all six directed number-format pairs.

The same descriptive pattern appears in an independent semantic domain (Fig. 1b). For musical pitch represented in scientific, lexical, and Helmholtz notation, held-out state transport reaches 0.802. Static update transport is only 0.083, whereas depth-evolving transport reaches 0.762, essentially matching the update-specific control at 0.759. The corresponding matched-sample gap recovery is 1.005, with the same pattern in all six directed notation pairs. A depth-consistent semantic shuffle eliminates the pitch dynamic alignment (approximately -0.001).

Thus the failure of a frozen map is not specific to number representations, and it does not imply that trajectory information is absent from the state correspondence. Instead, the relevant object can be the layer-indexed family $\{T_\ell\}$ rather than a single T_ℓ. Because both matched-sample analyses can share estimation error across adjacent maps, we next quantify how much of this recovery survives independent map estimation.

4.2 Recovery is driven by map evolution and survives independent estimation

For affine maps $T_\ell(x) = L_\ell x + c_\ell$, depth-evolving transport admits the exact decomposition

$$T_{\ell+1}(X^\ell + V^\ell) - T_\ell(X^\ell) = L_\ell V^\ell + (L_{\ell+1} - L_\ell)X^\ell + (L_{\ell+1} - L_\ell)V^\ell + (c_{\ell+1} - c_\ell).$$

The first term is the frozen-map prediction; the remaining terms arise from evolution of the state correspondence across the update.

Empirically, the state-dependent map-evolution term provides most of the additional alignment. Starting from 0.0963, adding $(L_{\ell+1} - L_\ell)X^\ell$ raises alignment to 0.6173, and including the remaining update-dependent term raises it to 0.6826. The combined map-evolution contribution aligns

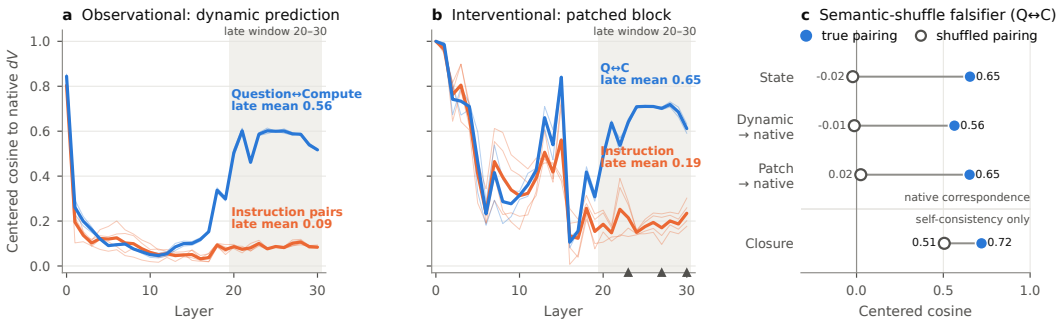

Figure 2: **A baseline-learned correspondence predicts and interventionally reproduces arithmetic-specific computation on held-out operands.** **(a)** Observational dynamic-to-native alignment across depth. Question–Compute correspondence emerges in late layers, averaging 0.56 over layers 20–30, while Instruction-involving pairs remain weak (0.09). Thick curves show group means and thin curves the directed formulation pairs. **(b)** Interventional test: after transporting the source state and executing the actual destination transformer block, the induced arithmetic-specific update aligns with the native destination update. Over layers 20–30, the Question–Compute mean is 0.65, compared with 0.19 for Instruction-involving pairs. **(c)** Semantic-shuffle falsifier for Question–Compute on the common evaluated layers $23, 27, 30$. Shuffling identity pairings collapses state correspondence, native trajectory alignment, and patch-to-native alignment to near zero, while substantial update-level square closure remains. Local dynamical self-consistency is therefore insufficient to identify the native computational correspondence.

with the target update at 0.667, whereas the static component aligns at 0.096; the two are nearly orthogonal on average (cosine -0.068).

The near-saturation under matched-sample fitting is nevertheless partly an estimation effect. With the same half of the training identities used for both adjacent maps, dynamic alignment is 0.634. Fitting the two maps on disjoint halves reduces it to 0.351, while a shuffled-pairing control reaches only 0.003. Under the same half-sample budget, the static baseline is 0.090 and the update-specific control is 0.643, so the cross-fitted dynamic predictor recovers

$$\frac{0.351 - 0.090}{0.643 - 0.090} = 0.471$$

of the finite-sample-matched gap (Fig. 1c). Across 50 random splits the cross-fitted dynamic score has an empirical 2.5–97.5 percentile range of $[0.328, 0.369]$. Shared estimation therefore contributes substantially to the matched-sample near-saturation, but does not explain the recovered trajectory signal: independently estimated adjacent correspondences retain nearly half of the finite-sample-matched static-to-update gap while shuffled pairings eliminate it.

4.3 Computational correspondence emerges on held-out arithmetic operands

We next ask whether a correspondence learned without arithmetic-specific trajectory information predicts computation on unseen operands. It does, but selectively across prompt formulations (Fig. 2a).

Question and Compute begin to develop strong bidirectional trajectory correspondence around layers 18–20. Over layers 20–30, their dynamic-to-native alignment averages 0.56. In contrast, pairs involving Instruction average only 0.09 over the same window.

Because the correspondence is fitted only on baseline training-operand states, this trajectory prediction is out of sample with respect to both held-out operand identity and nonzero-addition trajectories. Sharing the same arithmetic operation and answer is therefore not sufficient for all three surface formulations to exhibit the same trajectory correspondence.

The separation is not simply inherited from generic prompt-form differences: at layer 28, Compute–Instruction similarity falls from 0.833 for the corresponding +0 states to 0.103 for the

arithmetic-specific state displacement, and from 0.652 to 0.060 for the corresponding updates (Appendix C.4).

Importantly, Instruction-involving pairs are also weaker at the level of state correspondence. The present experiment therefore does not isolate a trajectory difference while holding endpoint alignment fixed; rather, it identifies a joint weakening of state and trajectory correspondence for the Instruction formulation.

4.4 Transported states induce corresponding updates under intervention

The observational correspondence survives a stronger interventional test (Fig. 2b). At each destination layer we transport the source state with the baseline-learned correspondence and then execute the actual destination transformer block. The induced arithmetic-specific update is compared with the native destination update.

Question–Compute again separates from Instruction-involving pairs in the same late-layer regime. Over layers 20–30, patch-to-native update alignment averages 0.65 for Question–Compute and 0.19 for pairs involving Instruction. Thus, for the Question and Compute formulations, the learned correspondence extends beyond observational trajectory prediction: transported states induce aligned arithmetic-specific updates under the native destination computation. This effect depends on the semantic correspondence rather than depth-consistent transport alone: the semantic-shuffle control below reduces patch-to-native alignment from approximately 0.65 to 0.02.

4.5 Local dynamical compatibility is insufficient for native correspondence

Local dynamical compatibility is weaker than correspondence with the native trajectory. The semantic-shuffle control makes the distinction explicit (Fig. 2c). On the common Question–Compute layers $23, 27, 30$, the true pairing gives approximately 0.65 state correspondence, 0.56 dynamic-to-native alignment, 0.65 patch-to-native alignment, and 0.72 update-level square closure. After shuffling semantic identity pairings, the first three quantities collapse to approximately -0.02, -0.01, and 0.02, respectively, whereas square closure remains substantial at approximately 0.51.

A semantically incorrect correspondence can therefore remain dynamically self-consistent while losing correspondence with the native states, trajectory, and induced update. Approximate layerwise closure is not sufficient to identify the intended computational correspondence.

4.6 Robustness and scope

The failure of frozen representation-derived update transport is robust across Pythia-1B, Pythia-1.4B, and Llama-3.1-8B, both pooling choices, and the tested rank sweep (Appendix D.1). Musical pitch additionally shows that the matched-sample static-to-dynamic recovery is not specific to numerical surface forms.

The stronger tests remain deliberately narrower. The adjacent-map decomposition and independent cross-fit audit are performed in the number domain with Llama-3.1-8B, and the out-of-sample arithmetic and intervention experiments use controlled addition in the same model. We therefore treat the cross-model results as evidence for the robustness of frozen-map failure, the pitch experiment as an independent-domain replication of matched-sample dynamic recovery, and the number/arithmetic experiments as the primary tests of estimation robustness and computational correspondence.

5 Limitations

Our experiments deliberately use structured domains with known identity correspondences. Matched-sample depth-evolving recovery appears in both number and musical-pitch representations, providing evidence that the phenomenon is not specific to a single semantic variable. However, the decomposition and independent adjacent-map audit are performed in the number domain, and the active-computation experiments are restricted to controlled addition and three prompt formulations in Llama-3.1-8B. Whether comparable depth-dependent correspondences arise for less structured natural-language concepts, other computations, and other model families remains to be established.

The arithmetic intervention also does not establish that the transported state is uniquely effective among perturbations of comparable magnitude. Because update similarity can increase simply when a perturbed state remains close to the native destination state, a stronger control would compare the transported intervention with random or structured perturbations matched to its state-space error norm. We leave this comparison to future work.

The three arithmetic formulations differ in surface features beyond question versus instruction style, including operand order and answer cues. We therefore do not identify which individual surface factor drives the weaker state and trajectory correspondence involving Instruction. Controlled template perturbations would be required to separate these factors.

The learned correspondences are finite-sample statistical objects. Adjacent maps fitted on the same examples can share estimation error: in the number audit, independent cross-fitting reduces dynamic alignment from 0.634 to 0.351, although the latter remains far above the shuffled control at 0.003 and recovers 47.1% of the finite-sample-matched static-to-update-specific gap. The matched-sample near-saturation in either numbers or pitch should therefore not be interpreted as evidence that the underlying correspondence is recovered without estimation uncertainty. More generally, differences between adjacent fitted maps can mix genuine depth evolution with finite-sample variation and non-identifiability of the low-rank correspondence.

6 Conclusion

A correspondence between representations at one depth does not generally specify a correspondence between the computations that move those representations. Across both number and musical-pitch domains, a frozen state map transports local updates poorly, while the evolution of state correspondences across adjacent depths recovers much of the corresponding trajectory. In numbers, this recovery is driven primarily by map evolution and remains substantial under independent adjacent-map estimation.

Controlled arithmetic shows that the same idea can extend beyond descriptive geometry. Correspondences learned only from baseline states predict arithmetic-specific trajectories for held-out operands, and transported Question–Compute states induce aligned updates when executed by the actual destination transformer block. At the same time, semantic shuffling shows that local dynamical square closure can survive after native state, trajectory, and interventional correspondence have disappeared.

These results motivate treating computational correspondence as a trajectory-level property of a depth-indexed family of maps, rather than as a consequence of state alignment at a single layer. The relevant test is not only whether two representations can be aligned, but whether their correspondence predicts the native computation that carries one state to the next.

Reproducibility Statement

All geometric objects are fit on fixed training identities and evaluated on held-out identities shared across surface-form domains. The supplementary material provides full experimental configurations, prompts, analysis details, random seeds, and code for activation extraction, geometry fitting, causal interventions, statistical analyses, and figure generation.

References

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

Taco S. Cohen, Mario Geiger, and Maurice Weiler. A general theory of equivariant CNNs on homogeneous spaces. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Wenyu Du, Rui Zheng, Tongxu Luo, Stephen Chung, and Jie Fu. Fine-grained manipulation of arithmetic neurons. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting*

Neural Networks for NLP, pp. 467–479. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.blackboxnlp-1.27.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 9574–9586, 2021.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah D. Goodman, and Christopher Potts. Inducing causal structure for interpretable neural networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 7324–7338. PMLR, 2022.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah D. Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236 of *Proceedings of Machine Learning Research*, pp. 160–187. PMLR, 2024.

Wes Gurnee and Max Tegmark. Language models represent space and time. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.02207.

Michael Hanna, Ollie Liu, and Alexandre Variengien. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Zhimin Hu, Lanhao Niu, and Sashank Varma. Representational geometry reveals how context structures concept spaces in language models, 2026. URL https://openreview.net/forum?id=wg6Q5fPL1m. ICML 2026 Mechanistic Interpretability Workshop Spotlight.

Marek Kadlčík, Michal Štefánik, Timothee Mickus, Michal Spiegel, and Josef Kuchař. Pre-trained language models learn remarkably accurate representations of numbers. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.emnlp-main.1356/.

Amit Arnold Levy and Mor Geva. Language models encode numbers using digit representations in base 10. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 385–395, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.naacl-short.33/.

Alex Modell. The origins of representation manifolds in large language models. *arXiv preprint arXiv:2505.18235*, 2025.

Sharath Naganna, Tanvir Ahmed Sijan, and Uddipta Kalita. Are arithmetic heuristic neurons form-invariant? a mechanistic analysis of symbols, text, and code in LLMs. *arXiv preprint arXiv:2607.16693*, 2026.

Gregory Polyakov, Christian Hepting, Carsten Eickhoff, and Seyed Ali Bahrainian. Interpretability analysis of arithmetic in-context learning in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1758–1777. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.92.

Philip Quirke and Fazl Barez. Understanding addition in transformers. In *International Conference on Learning Representations*, 2024.

Or Shafran, Shaked Ronen, Omri Fahn, Shauli Ravfogel, Atticus Geiger, and Mor Geva. From directions to regions: Decomposing activations in language models via local geometry, 2026. URL https://arxiv.org/abs/2602.02464.

Michal Štefánik, Timothee Mickus, Marek Kadlčík, Bertram Højer, Michal Spiegel, Raúl Vázquez, Aman Sinha, Josef Kuchař, Philipp Mondorf, and Pontus Stenetorp. Language models learn universal representations of numbers and here's why you should care. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, 2026. URL https:

//aclanthology.org/2026.acl-long.1415/. Also presented at the ICML 2026 Mechanistic Interpretability Workshop.

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

Yao Yan. Addition in four movements: Mapping layer-wise information trajectories in LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 7518–7532. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.397.

Long Zhang, Dai-jun Lin, and Wei-neng Chen. The geometric price of discrete logic: Context-driven manifold dynamics of number representations. *arXiv preprint arXiv:2603.23577*, 2026.

A Additional Related Work

Representation manifolds, vector fields, and depth evolution. Prior work has described concepts using low-dimensional directions, subspaces, and manifolds in activation space (Gurnee & Tegmark, 2024; Shafran et al., 2026; Modell, 2025). Most closely related to our geometric framing, Hu et al. (2026) model concepts as point-cloud manifolds and contextual transformations as vector fields, while Zhang et al. (2026) studies task-dependent deformation of number manifolds during reasoning. Our object differs in two respects. First, we study the internal block-induced displacement

$$V^\ell = h^{\ell+1} - h^\ell,$$

rather than a displacement generated by changing external context. Second, our central object is not an individual state correspondence T_ℓ but its evolution across depth. The transported displacement is therefore

$$T_{\ell+1}(h^{\ell+1}) - T_\ell(h^\ell),$$

which need not equal the static linear transport $L_\ell V^\ell$ even when T_ℓ accurately aligns the states.

Equivariance, intertwining, and computational correspondence. Commuting relations between transformations and learned maps are central to equivariant neural networks, where layers are constructed to intertwine known group representations (Cohen et al., 2019). Our relation

$$F_B^\ell \circ T_\ell \approx T_{\ell+1} \circ F_A^\ell$$

has a similar mathematical form but a different empirical role. The maps T_ℓ are estimated from matched hidden states rather than supplied by a known group action, and the transformer blocks are not constrained to satisfy the relation. We therefore use approximate square closure as a diagnostic of local dynamical compatibility rather than as an architectural symmetry. Crucially, our shuffled-pairing experiment separates this compatibility from semantic correctness: a depth-consistent but semantically incorrect correspondence can retain substantial square closure while failing to align native states and trajectories.

Causal abstraction and interchange interventions. Causal-abstraction methods formalize when internal variables of a neural network realize variables and mechanisms of a higher-level causal model (Geiger et al., 2021; 2022). Interchange interventions test such proposed alignments by replacing aligned internal representations and comparing the resulting counterfactual behavior, while distributed alignment search extends the framework to alignments in distributed representational subspaces (Geiger et al., 2024). Our intervention is related in spirit but addresses a different correspondence. Rather than testing whether a neural representation realizes a predefined high-level causal variable, we transport a state between two surface-form realizations within the same network and execute the actual destination transformer block. The question is whether a state correspondence learned without the target computation extends to the update induced by that computation.

Arithmetic representations, mechanisms, and trajectories. Arithmetic information has been studied at several complementary levels. Operands, intermediate quantities, and answers can be decoded from hidden states (Yan, 2025; Sun et al., 2025); mechanistic studies have localized arithmetic behavior to circuits, components, function vectors, or sparse heuristic neurons (Hanna et al., 2023; Quirke & Barez, 2024; Polyakov et al., 2025; Du et al., 2025); and recent work examines whether such mechanisms persist across different surface forms (Naganna et al., 2026). These questions differ from trajectory correspondence. Recoverability of an arithmetic quantity establishes structure in state space, while component overlap identifies where a computation may be implemented. Neither alone establishes that a correspondence between two formulations predicts how their states move from layer to layer. Our arithmetic experiments test this latter property out of sample by fitting correspondences only on baseline states of training operands and evaluating arithmetic-specific trajectories on unseen operands.

B Experimental and Methodological Details

This section provides the experimental details omitted from the main text, including prompt templates, layer conventions, causal intervention construction, control conditions, and uncertainty estimation.

B.1 Number prompts and data splits

The controlled number experiments use values 0–99 represented in three surface-form domains: Arabic digits, English number words, and ordinals. For digits and number words, representations are extracted using

```
The number is {x}.
I wrote the number {x}.
The sequence contains {x}.,
```

while ordinal prompts use

```
The ordinal is {x}.
The ranking is {x}.
The item came {x}..
```

For each domain and identity, hidden states are pooled according to the selected pooling rule and then averaged across the three templates before fitting representation or update geometry.

We use a fixed split of 70 training and 30 held-out identities, shared across all surface-form domains. Means, PCA bases, and transport maps are estimated using training identities only; no representation of a held-out identity contributes to a fitted geometric object.

B.2 Static-geometry diagnostics

The main text focuses on the distinction between a frozen state map and the depth-evolving family of state correspondences. Here we define additional diagnostics used to characterize why the frozen-map prediction

$$\widehat{V}^{\ell}_{B,\mathrm{static}} = V^{\ell}_{A} P^{A\to B}_{X,\ell}$$

can fail. These diagnostics concern only the geometry of a single layer and should not be interpreted as evidence that state and update correspondences obey intrinsically different transformation laws.

For each domain A and layer ℓ, let

$$\bar{X}^{\ell}_{A}(c) = X^{\ell}_{A}(c) - \mu^{\ell}_{X,A}, \qquad \bar{V}^{\ell}_{A}(c) = V^{\ell}_{A}(c) - \mu^{\ell}_{V,A}$$

denote states and updates centered using training identities. Let $B^{\ell}_{X,A}$ and $B^{\ell}_{V,A}$ denote rank-r PCA bases fitted separately to the centered state and update clouds.

Update energy inside the state-derived support. We first measure how much of a held-out update lies inside the subspace identified from representations. Writing

$$\Pi^{\ell}_{X,A} = B^{\ell}_{X,A}(B^{\ell}_{X,A})^{\top},$$

we define

$$E^{A,\ell}_{V|X} = \mathbb{E}_{c\in\mathcal{T}_{\text{test}}}\left[\frac{\left\|\bar{V}^{\ell}_{A}(c)\Pi^{\ell}_{X,A}\right\|_2^2}{\left\|\bar{V}^{\ell}_{A}(c)\right\|_2^2}\right]. \tag{14}$$

We compute the same quantity in the destination domain. Because any prediction restricted to the state-derived support is orthogonal to the residual component outside that support, the corresponding per-example geometric ceiling on cosine alignment is

$$C^{A,\ell}_{\max,X} = \mathbb{E}_{c\in\mathcal{T}_{\text{test}}}\left[\frac{\left\|\bar{V}^{\ell}_{A}(c)\Pi^{\ell}_{X,A}\right\|_2}{\left\|\bar{V}^{\ell}_{A}(c)\right\|_2}\right]. \tag{15}$$

This diagnostic separates lack of update support from errors in the fitted cross-domain transformation.

Update refit within the state-derived support. To test whether the frozen state map fails only because its Procrustes alignment was estimated from states rather than updates, we hold the state-derived source and destination bases fixed but refit the orthogonal map using update coordinates:

$$R^{A\to B}_{V|X,\ell} = \mathrm{Proc}\left(\bar{V}^{\ell}_{A}B^{\ell}_{X,A},\, \bar{V}^{\ell}_{B}B^{\ell}_{X,B}\right), \tag{16}$$

with

$$P^{A\to B}_{V|X,\ell} = B^{\ell}_{X,A}R^{A\to B}_{V|X,\ell}(B^{\ell}_{X,B})^{\top}. \tag{17}$$

The held-out prediction

$$\widehat{V}^{\ell}_{B,V|X} = \bar{V}^{\ell}_{A}P^{A\to B}_{V|X,\ell}$$

therefore uses exactly the same source and destination support as the representation-derived predictor. Improvement over $\bar{V}^{\ell}_{A}P^{A\to B}_{X,\ell}$ indicates that transformation mismatch exists even after support is held fixed; failure of both predictors indicates that support mismatch itself is an important limitation.

Matched union support. The representation and update PCA bases need not span the same directions. To compare their fitted transport laws without giving either one a distinct support, we construct for each domain a common rank-r basis from normalized state and update clouds,

$$B^{\ell}_{U,A} = \mathrm{PCA}_r\left(\begin{bmatrix}\bar{X}^{\ell}_{A}/\|\bar{X}^{\ell}_{A}\|_F \\ \bar{V}^{\ell}_{A}/\|\bar{V}^{\ell}_{A}\|_F\end{bmatrix}\right), \tag{18}$$

and analogously $B^{\ell}_{U,B}$. The Frobenius normalization prevents the larger of the two clouds from determining the union basis solely through its overall scale.

Within this shared support, we fit one orthogonal map from states,

$$R^{A\to B}_{X|U,\ell} = \mathrm{Proc}\left(\bar{X}^{\ell}_{A}B^{\ell}_{U,A},\, \bar{X}^{\ell}_{B}B^{\ell}_{U,B}\right), \tag{19}$$

and one from updates,

$$R^{A\to B}_{V|U,\ell} = \mathrm{Proc}\left(\bar{V}^{\ell}_{A}B^{\ell}_{U,A},\, \bar{V}^{\ell}_{B}B^{\ell}_{U,B}\right). \tag{20}$$

The corresponding ambient-space maps are

$$P^{A\to B}_{X|U,\ell} = B^{\ell}_{U,A}R^{A\to B}_{X|U,\ell}(B^{\ell}_{U,B})^{\top}, \tag{21}$$

$$P^{A\to B}_{V|U,\ell} = B^{\ell}_{U,A}R^{A\to B}_{V|U,\ell}(B^{\ell}_{U,B})^{\top}. \tag{22}$$

Both maps therefore act on exactly the same source coordinates and predict updates inside exactly the same destination support. Their held-out difference,

$$\Delta C^{\ell}_{U} = C\left(\bar{V}^{\ell}_{A}P^{A\to B}_{V|U,\ell}, \bar{V}^{\ell}_{B}\right) - C\left(\bar{V}^{\ell}_{A}P^{A\to B}_{X|U,\ell}, \bar{V}^{\ell}_{B}\right), \tag{23}$$

isolates the difference between state-fitted and update-fitted static transport after support and rank are matched.

These quantities diagnose the failure of the *frozen* predictor. In particular, low $E_{V|X}$ establishes that the state-derived support alone cannot capture most update energy, while the same-support and union-support comparisons test whether an additional mismatch remains in the fitted static map. None of these diagnostics includes the adjacent-depth term $T_{\ell+1}(X^{\ell+1}_{A}) - T_{\ell}(X^{\ell}_{A})$; they are therefore complementary to, rather than alternatives to, the depth-evolving analysis in the main text.

B.3 STATIC-TRANSPORT INTERVENTION

As a secondary diagnostic, we test the functional consequences of replacing the within-support component of a target update using either the static representation-derived or update-derived transport law. This experiment tests static transport only and is distinct from the arithmetic block intervention in the main text.

For a held-out target update, write

$$\bar{V}_B(c) = u_{\text{native}}(c) + r_\perp(c), \tag{24}$$

where u_{native} is the centered component in the shared union support of Appendix B.2 and $r_\perp$ is the complete orthogonal residual. Using the corresponding source update, let u_X and u_V denote transport by the representation-fit and update-fit union maps, respectively. We construct

$$V_{B,j}(c) = \mu_{V,B} + r_\perp(c) + u_j(c), \qquad j \in \{X, V\}. \tag{25}$$

Thus the target mean, off-union residual, source coordinate, support, and rank are matched between conditions; only the fitted static transport law changes. We additionally evaluate a norm-matched condition in which the transported component is rescaled to the norm of the native target union component.

Let p_{clean} denote the clean next-token distribution and define

$$D_X = D_{\text{KL}}(p_{\text{clean}} \| p_X), \qquad D_V = D_{\text{KL}}(p_{\text{clean}} \| p_V), \tag{26}$$

with matched-map contrast

$$\Delta_{\text{KL}} = D_X - D_V. \tag{27}$$

A deletion control removes the centered within-union component, while a semantic-shuffle control applies update-derived transport after breaking the source–target identity pairing.

This intervention should be interpreted as a diagnostic of the inadequacy of *single-layer* representation-derived update transport. Because the main results show that a depth-evolving state correspondence contains additional trajectory information omitted by $L_\ell V^\ell$, the experiment does not compare the complete state-derived trajectory predictor with the update-specific transport.

B.4 MODELS, POOLING, AND LAYER CONVENTIONS

Experiments use Pythia-1B, Pythia-1.4B, and Llama-3.1-8B. Hidden states are extracted either at the final token of the target span or by averaging over the target-token span.

For a model with L transformer blocks, normalized depth is

$$d_\ell = \frac{\ell}{L-1},$$

and aggregate number analyses summarize middle layers satisfying $0.2 \le d_\ell \le 0.8$.

For Llama-3.1-8B, block-update analyses use $\ell = 0, \ldots, 30$. We exclude the final decoder-block update when updates are constructed from HuggingFace hidden-state differences because the terminal hidden state includes the model's final normalization and is therefore not directly comparable to the preceding block-output differences.

B.5 CAUSAL INTERVENTION CONSTRUCTION

Residual-stream interventions use three neutral target prompt frames distinct from the templates used to estimate the geometric maps:

```
The number is {x}
I wrote {x}
The value is {x}.
```

Representation- and update-derived transport therefore face the same prompt shift.

For a held-out target update, write

$$\bar{V}_B(c) = u_{\text{native}}(c) + r_\perp(c),$$

where u_{native} is the centered component in the shared union support and $r_\perp$ is its complete orthogonal residual. The transported components are

$$u_X(c) = \bar{V}_A(c) B_{U,A} R_{X|U} B_{U,B}^\top, \tag{28}$$

$$u_V(c) = \bar{V}_A(c) B_{U,A} R_{V|U} B_{U,B}^\top. \tag{29}$$

The corresponding patched updates are

$$V_{B,X}(c) = \mu_{V,B} + r_\perp(c) + u_X(c), \tag{30}$$

$$V_{B,V}(c) = \mu_{V,B} + r_\perp(c) + u_V(c). \tag{31}$$

Thus the target prompt, target mean update, off-union residual, source coordinate, support, and rank are identical between the two transport conditions.

Because both transport maps are orthogonal and act on the same source coordinate,

$$\|u_X(c)\|_2 = \|u_V(c)\|_2.$$

We additionally evaluate a norm-matched condition in which transported components are rescaled to the norm of the native target union component. The norm-matched condition is used for the static-intervention comparison reported below.

B.6 Mean-pooled causal patching

For a target-token span S, let

$$\bar{v}_{\text{native}} = \frac{1}{|S|} \sum_{j \in S} (h_j^{\ell+1} - h_j^\ell)$$

denote its native mean update, and let $\bar{v}^\star$ denote the desired patched mean update. We apply the common displacement

$$\delta = \bar{v}^\star - \bar{v}_{\text{native}}$$

to every token in the target span:

$$h_j^{\ell+1} \leftarrow h_j^{\ell+1} + \delta, \qquad j \in S.$$

This exactly realizes the desired mean-pooled update while preserving within-span token-to-token deviations. Final-token intervention is the special case $|S| = 1$.

Both pooling choices yield the same qualitative causal separation. For span-mean pooling, the norm-matched contrast is

$$\Delta_{\text{KL}} = D_X - D_V = 0.00441, \qquad 95\%\ \text{CI} = [0.00366, 0.00524],$$

with positive confidence intervals for all six directed domain pairs.

B.7 Causal controls and sanity checks

Let p_{clean} denote the clean next-token distribution and define

$$D_X = D_{\text{KL}}(p_{\text{clean}} \| p_X), \qquad D_V = D_{\text{KL}}(p_{\text{clean}} \| p_V),$$

so that the primary matched-map contrast is

$$\Delta_{\text{KL}} = D_X - D_V.$$

The deletion condition removes only the centered component inside the tested union support:

$$V_{B,0}(c) = \mu_{V,B} + r_\perp(c).$$

Writing its divergence as D_0, the deletion-relative costs are

$$C_X = D_X - D_0, \qquad C_V = D_V - D_0.$$

Thus $C > 0$ indicates that inserting the transported component is more disruptive than deleting it, whereas $C < 0$ indicates an improvement relative to deletion.

We additionally evaluate update-derived transport after shuffling semantic source–target correspondences. This preserves the update-derived map family, source-coordinate distribution, and intervention construction while breaking the identity-specific semantic correspondence.

As an implementation sanity check, reconstructing the target update from its training-set mean, complete off-union residual, and native union component reproduces the clean forward pass to numerical precision. This verifies both the decomposition and the intervention hook.

B.8 Musical-pitch experimental details

The musical-pitch replication contains 60 identities: the 12 chromatic pitch classes across octaves 2–6. Each pitch is expressed in three notation systems: scientific notation, a digit-free lexical description, and ASCII Helmholtz-style notation. For example, the same pitch may appear as `F#4`, "F sharp in the fourth octave," or `f#'`.

Pitch identities are split $70/30$ into fitting and held-out sets, so no notation of a held-out pitch contributes to a fitted mean, basis, or map. We evaluate all six directed notation pairs in Llama-3.1-8B. The main-text pitch replication uses mean pooling and rank 32. For each update layer, we compute the frozen-map prediction and the depth-evolving endpoint prediction exactly as in the number experiment, together with an update-specific positive control. A semantic-shuffle control applies a fixed permutation of training pitch identities consistently across adjacent depths before fitting the state maps. Additional rank and pooling robustness analyses use $r \in \{4, 8, 16, 32\}$ and both final-token and mean pooling.

B.9 Active-arithmetic task construction and calibration

The active-arithmetic experiments use controlled addition under three semantically equivalent prompt formulations:

Question: `What is {n} plus {Δ}?  Answer:`,
Compute: `Compute {n} plus {Δ}.  Answer:`,
Instruction: `Add {Δ} to {n}.  Result:`.

We use offsets

$$\Delta \in \{1, 2, 5, 10, 20, 37\}.$$

For each offset, we include every operand $n \in \{0, \ldots, 99\}$ for which the result remains in the canonical number set, $n + \Delta \in \{0, \ldots, 99\}$. This gives

$$(100 - 1) + (100 - 2) + (100 - 5) + (100 - 10) + (100 - 20) + (100 - 37) = 525$$

nonzero-addition problems. For each formulation, the corresponding $\Delta = 0$ prompt provides a formulation-matched baseline.

Let $H_p^\ell(n, \Delta)$ denote the hidden state under formulation p, and define the block update

$$V_p^\ell(n, \Delta) = H_p^{\ell+1}(n, \Delta) - H_p^\ell(n, \Delta).$$

We isolate arithmetic-specific state displacements and differential updates by subtracting the matched $\Delta = 0$ condition:

$$E_p^\ell(n, \Delta) = H_p^\ell(n, \Delta) - H_p^\ell(n, 0), \qquad dV_p^\ell(n, \Delta) = V_p^\ell(n, \Delta) - V_p^\ell(n, 0).$$

This subtraction removes formulation-specific baseline structure without requiring a fitted cross-context map.

For comparison with neutral number geometry, we use the canonical digit representations $X^\ell(n)$ and define the operand-to-result displacement

$$D^\ell(n, \Delta) = X^\ell(n + \Delta) - X^\ell(n).$$

For endpoint decoding, numerical identities are randomly divided into 70% training and 30% test nodes over 20 splits (seed 271828). Training and test arithmetic edges require both n and $n + \Delta$ to

lie within the corresponding node partition, preventing numerical-identity overlap between the two sets. For each split and layer, we fit a rank-32 affine bridge using only the $\Delta = 0$ states of training identities,

$$C_\ell : H_p^\ell(n, 0) \mapsto X^\ell(n),$$

and apply the same map to nonzero-addition states on held-out test edges. No nonzero-addition state is used to fit the bridge.

As a bridge-free endpoint diagnostic, we independently compare the relational geometry of arithmetic states with the corresponding canonical operand and result geometries. Cross-formulation trajectory analyses compare matched problems after the formulation-specific $\Delta = 0$ subtraction above. Behavioral correctness is evaluated by greedy generation. Cross-formulation trajectory comparisons are restricted to the common-correct subset, i.e., problems answered correctly under all three formulations.

The supplementary endpoint and direct pairwise trajectory diagnostics in Appendix C use layers 23–30 of Llama-3.1-8B unless otherwise stated. The main cross-formulation transport and intervention figures report the full layerwise curves and summarize the late window over layers 20–30; the semantic-shuffle falsifier uses the common evaluated layers 23, 27, and 30.

B.10 Uncertainty estimation

For number transport rank sweeps, curves average over the six directed surface-form pairs after averaging across middle layers. Error bars show the standard error across directed pairs and are interpreted descriptively rather than as independent-sample hypothesis tests.

For causal analyses, uncertainty is estimated by bootstrap resampling held-out numerical identities. Repeated layers, templates, and directed pairs belonging to the same identity are averaged before resampling. Musical-pitch confidence intervals analogously bootstrap held-out pitch identities.

For arithmetic endpoint–trajectory correspondence, significance is assessed using the semantic-permutation null described in Appendix C.2. Confidence intervals use an operand-cluster bootstrap that resamples operands while retaining all available offsets and analyzed layers associated with each operand. A problem-level bootstrap provides an additional robustness check.

C Additional Arithmetic Analyses

The main arithmetic results test whether correspondences fitted only from baseline states predict arithmetic-specific trajectories on held-out operands, whether transported states induce corresponding updates under the destination network, and whether these effects survive controls for semantic correctness of the correspondence. This section reports complementary diagnostics that place those results in the broader geometry of the arithmetic task.

In particular, we ask whether arithmetic-specific motion simply follows the canonical operand-to-result direction in neutral number representations, whether the three prompt formulations differ in endpoint organization, and whether the Question–Compute versus Instruction hierarchy is also visible in direct pairwise trajectory comparisons. These diagnostics are not used to define computational correspondence in the main text; rather, they provide independent geometric and statistical context for the transport, intervention, and semantic-shuffle results reported there.

Unless otherwise stated, the late-layer diagnostics below use layers 23–30 of Llama-3.1-8B. Direct cross-formulation comparisons use the 517 arithmetic problems answered correctly under all three formulations.

C.1 Additional endpoint and route diagnostics

For formulation p, define the arithmetic-specific state displacement

$$E_p^\ell(n, \Delta) = H_p^\ell(n, \Delta) - H_p^\ell(n, 0), \tag{32}$$

and the canonical operand-to-result displacement in neutral number geometry,

$$D^\ell(n, \Delta) = X^\ell(n + \Delta) - X^\ell(n). \tag{33}$$

The cosine between $dV_p^\ell(n, \Delta)$ and $D^\ell(n, \Delta)$ measures a different property from cross-formulation transport: it asks whether the local arithmetic-specific block update points along the displacement that connects the neutral representations of the operand and result.

We also characterize endpoint organization using a train-only affine bridge from formulation-specific baseline states to canonical number geometry. Numerical identities are split into disjoint training and test sets, and the bridge is fitted only from $\Delta = 0$ states of training identities before being applied to nonzero-addition states on held-out arithmetic examples. No nonzero-addition state is used to fit the bridge.

As a bridge-independent endpoint diagnostic, we compare the relational geometry of arithmetic states directly with canonical operand and result geometries and define

$$M_p^\ell = \rho_{p,\mathrm{result}}^\ell - \rho_{p,\mathrm{operand}}^\ell. \tag{34}$$

Positive values indicate a relatively more result-like organization and negative values a relatively more operand-like organization.

Finally, direct matched-problem similarities of E and dV across prompt formulations can be summarized by the Question–Compute contrast

$$G = S_{QC} - \frac{1}{2}(S_{QI} + S_{CI}). \tag{35}$$

These quantities characterize endpoint organization and direct pairwise route similarity. They are complementary to the main transport analysis, which asks a stricter out-of-sample question: whether a correspondence learned only from baseline states of training operands predicts the arithmetic-specific trajectory of unseen operands.

C.2 Local updates are only weakly aligned with the canonical operand-to-result displacement

We first ask whether arithmetic-specific local updates can be identified with the neutral representational displacement from operand to result. Across layers 23–30, the matched mean cosine between $dV^\ell(n, \Delta)$ and $D^\ell(n, \Delta)$ is only 0.00862.

To determine whether this small value nevertheless contains identity-specific structure, we construct a semantic-permutation null that preserves layer and offset while shuffling numerical identity. The null mean is 0.00022, giving a matched-minus-shuffled excess of

$$0.00840,$$

with $p_{\mathrm{perm}} < 0.001$.

The estimate is stable under two resampling schemes. A problem-level bootstrap over the 525 operand–offset pairs gives

$$95\%\ \mathrm{CI} = [0.00790,\ 0.00892],$$

while an operand-cluster bootstrap, which retains all offsets associated with each resampled operand, gives

$$95\%\ \mathrm{CI} = [0.00767,\ 0.00914].$$

Thus local arithmetic motion contains a statistically detectable identity-specific component along the canonical operand-to-result displacement, but the effect is extremely small in absolute magnitude.

As a descriptive layerwise diagnostic, correct-result recovery in canonical number geometry peaks near layer 28, whereas the matched-minus-shuffled local update alignment at that layer is -0.00602; the corresponding values at layers 27 and 29 are 0.01731 and 0.01766. Because this contrast was not specified independently of the aggregate analysis, we do not treat it as a separate statistical result. The diagnostic instead illustrates that recovery of a result-like endpoint need not imply that an individual transformer block moves directly along the neutral operand-to-result vector.

This result is conceptually distinct from the main cross-formulation trajectory experiment. The latter tests whether a learned correspondence maps the native trajectory of one prompt formulation to that of another; it does not require either trajectory to follow $D^\ell(n, \Delta)$.

C.3 Instruction endpoint organization is not a bridge-calibration artifact

The Instruction formulation shows weak canonical-result recovery under the affine bridge. We therefore test whether this effect can be explained simply by a failure to align Instruction states with canonical number geometry.

For

$$\texttt{Add } \{\Delta\} \texttt{ to } \{\texttt{n}\}.\ \ \texttt{Result:},$$

canonical-result retrieval remains approximately 2–3% across the analyzed late layers despite $517/525$ correct responses. Under the corresponding $+0$ formulation, however, the same calibration procedure retrieves the held-out operand at approximately 53–69% across layers 23–30. The bridge therefore retains substantial access to numerical identity in Instruction states even though it does not produce the strong result-recovery signature observed for Question and Compute.

More importantly, the endpoint difference persists without any fitted bridge. We compare the pairwise relational geometry of arithmetic states directly with canonical operand and result geometries using RSA. Averaged over layers 23–30, the result-minus-operand margins are

$$\begin{aligned} M_{\text{Question}} &= 0.081, \quad 95\%\ \text{CI} = [0.055, 0.106], \\ M_{\text{Compute}} &= 0.052, \quad 95\%\ \text{CI} = [0.023, 0.076], \\ M_{\text{Instruction}} &= -0.139, \quad 95\%\ \text{CI} = [-0.173, -0.101]. \end{aligned}$$

The corresponding prompt-family contrast,

$$\frac{1}{2}\left(M_{\text{Question}} + M_{\text{Compute}}\right) - M_{\text{Instruction}},$$

is 0.206 with 95% confidence interval $[0.157, 0.251]$.

Instruction still contains substantial numerical relational structure: its mean operand and result RSA values are 0.614 and 0.475, respectively. Its negative margin therefore indicates a relatively more operand-like organization, rather than an absence of numerical geometry.

These endpoint diagnostics are consistent with, but do not explain, the main trajectory result. In particular, the weaker Instruction trajectory correspondence occurs together with weaker state correspondence, so the present experiments do not isolate a trajectory difference while holding endpoint alignment fixed.

C.4 Direct route separation survives removal of the $+0$ baseline

The main analysis establishes the Question–Compute versus Instruction-involving hierarchy using correspondences learned from baseline training operands. As an independent diagnostic, we ask whether the same hierarchy is visible directly in arithmetic-specific state displacements and updates, without fitting a cross-formulation map.

At layer 28, pairwise cosine similarities between arithmetic-specific state displacements E are

$$S_{QC} = 0.802, \qquad S_{QI} = 0.122, \qquad S_{CI} = 0.103,$$

and the corresponding differential-update similarities are

$$S_{QC} = 0.751, \qquad S_{QI} = 0.054, \qquad S_{CI} = 0.060.$$

Thus the same qualitative hierarchy is visible directly: Question and Compute are strongly aligned, whereas both Instruction-containing comparisons are much weaker.

This separation is not inherited simply from generic differences between the prompt templates. At layer 28, Compute–Instruction similarity is 0.833 for the corresponding $+0$ states but falls to 0.103 for the arithmetic-specific displacement E. For updates, the corresponding similarity changes from 0.652 for the $+0$ updates to 0.060 for dV after subtraction of the formulation-specific baseline. The route-family separation therefore emerges in the arithmetic-specific component rather than merely reflecting the baseline geometry of the prompt formulations.

These direct similarities should not be confused with the main dynamic-to-native transport score. They show that the native arithmetic-specific trajectories themselves form the same qualitative prompt-family structure that the baseline-learned correspondence recovers out of sample.

C.5 Direct semantic-shuffle controls retain weak cross-family correspondence

The low Question–Instruction and Compute–Instruction direct similarities could reflect either weak but identity-specific correspondence or a complete loss of problem-specific structure. We distinguish these possibilities by shuffling numerical identities while preserving prompt pair and analysis layer.

Matched-problem similarity exceeds this semantic-shuffle baseline by

Prompt pair	E	dV
Question–Compute	0.537	0.613
Question–Instruction	0.031	0.036
Compute–Instruction	0.032	0.035

for arithmetic-specific state displacements and differential updates, respectively.

The Instruction-containing comparisons therefore retain a weak identity-specific correspondence above the shuffled baseline, but this correspondence is far smaller than for Question–Compute. We accordingly describe these conditions as distinct trajectory families, not as statistically independent trajectories or evidence for distinct arithmetic algorithms.

This control is complementary to the semantic-shuffle falsifier in the main text. Here the shuffle is applied when measuring *direct* matched-problem similarity of E and dV. In the main experiment, semantic identities are instead shuffled when fitting the cross-formulation state correspondence. That stronger control asks whether an internally depth-consistent but semantically incorrect map can still align native states, predict native trajectories, or induce native updates. The two controls therefore address different questions.

D Additional Robustness Analyses

This section tests whether the failure of frozen representation-derived update transport depends on model choice, pooling rule, or the use of global rather than local representation subspaces.

D.1 Cross-model and pooling robustness

Frozen-map failure is consistent across all three tested models and both pooling rules. State transport remains high, whereas applying the representation-derived correspondence statically to held-out updates performs poorly. Update-specific static transport remains substantially stronger, and only a small fraction of update energy lies in the representation-derived support. These results concern the frozen single-layer predictor; they do not test the depth-evolving transport analyzed in the main text.

Table 1: Robustness of frozen representation-derived update transport across number experiments. Entries average over the designated middle-depth window and directed surface-form pairs.

Model	Pool	Rep.	Update via rep.	Update fit	$E_{V\|X}$
Pythia-1B	last	0.956	0.095	0.661	0.084
Pythia-1B	mean	0.967	0.083	0.688	0.070
Pythia-1.4B	last	0.975	0.068	0.704	0.072
Pythia-1.4B	mean	0.978	0.061	0.705	0.065
Llama-3.1-8B	last	0.957	0.098	0.715	0.054
Llama-3.1-8B	mean	0.954	0.096	0.692	0.054

D.2 Local tangent alignment of numerical updates

The main experiments compare global representation and update subspaces and their cross-domain transport laws. As a complementary local diagnostic, we ask whether numerical updates preferentially follow tangent directions estimated from the local number representation manifold.

Table 2: **Local tangent rank sweep across models and pooling rules.** Entries average equally over Pythia-1B, Pythia-1.4B, and Llama-3.1-8B under final-token and mean pooling, as well as surface-form domains, valid numerical centers, and middle layers. The tangent/chance ratio is computed per condition relative to the corresponding isotropic baseline k/d before aggregation. Local tangent directions are enriched over ambient chance but capture only a small absolute fraction of update energy.

Rank k	Tangent fraction	Normal fraction	Tangent/chance
1	0.0036	0.9964	8.73
2	0.0055	0.9945	6.75
4	0.0081	0.9919	4.95
8	0.0119	0.9881	3.69

For each numerical identity n, we define a fixed value-space neighborhood with radius four,

$$\mathcal{N}(n) = \{n-4, \ldots, n, \ldots, n+4\}.$$

We retain only identities for which the complete nine-point neighborhood is available, so all tangent ranks are evaluated on the same set of centers. At each layer and within each surface-form domain, we apply PCA to the nine representations in $\mathcal{N}(n)$. Because the neighborhood is centered, its maximum possible PCA rank is eight. We compute a single rank-8 local basis and evaluate its leading $k \in \{1, 2, 4, 8\}$ directions.

For the native layer update

$$v^\ell(n) = h^{\ell+1}(n) - h^\ell(n),$$

we measure the fraction of update energy captured by the rank-k local tangent basis $B_k^\ell(n)$:

$$E_{\tan}^\ell(n; k) = \frac{\left\| v^\ell(n) B_k^\ell(n) B_k^\ell(n)^\top \right\|_2^2}{\|v^\ell(n)\|_2^2}.$$

An isotropic k-dimensional subspace in ambient dimension d has expected energy fraction k/d, which we use as the ambient chance baseline.

The local tangent component is enriched relative to the isotropic baseline at every tested rank, but remains small in absolute terms: the captured update energy increases from 0.0036 at rank 1 to only 0.0119 at rank 8. Numerical updates therefore exhibit a weak preference for locally number-aligned directions without being concentrated in the local representation tangent space.

This analysis is complementary to the main trajectory experiments. It tests local alignment with the numerical representation manifold rather than cross-domain trajectory transport. The weak local enrichment is therefore consistent with the narrower conclusion that a frozen single-layer representation geometry does not by itself determine the support or static transport of local updates.