CAN WE FIND THE OPTIMAL VARIABLE ORDERING FOR MAMBA IN LARGE-SCALE SYSTEMS?

Anonymous authors
Paper under double-blind review

ABSTRACT

When Mamba reads variable histories one by one, which variables should it read together? Large physical systems contain many pieces of equipment, each described by different signals. A tank's pressure, level, and flow, for example, measure different aspects of the same process. We hypothesize that keeping such heterogeneous signals together can improve prediction. We test a policy that groups signals by component while randomizing the sequence of groups. In a simulated nuclear power system, this policy gives lower mean test error than unrestricted random orders and a history-similarity order derived from GPS-Mamba. We then investigate the learned computation in a diagnostic comparison of a fixed component order with random orders. Removing an estimated output interaction between a target history and the other histories reverses the fixed order's advantage on shared diagnostic windows. An internal control, however, does not support attributing that advantage to the immediately preceding variable's write. Together, these experiments connect variable organization to prediction and distinguish joint use of histories from a simple local-write explanation.

1 INTRODUCTION

Mamba carries information through a state that it updates as it reads a sequence (Gu & Dao, 2024). For multivariate forecasting, a token can contain the observation history of one variable (Wang et al., 2024). The predictor then reads a sequence of variables. This makes variable order a modeling choice: which histories should enter the state next to one another?

The choice becomes difficult when many signals describe different parts of a coupled physical system. In a nuclear plant, water level, pressure, flow, temperature, and power each report a different aspect of operation. Standardizing the values does not change what each variable measures. Signals from one pressure-control tank may be useful together even when their time-series shapes differ. Plant documentation tells us which signals belong to that tank before we train a predictor.

We hypothesize that keeping a component's heterogeneous signals together can improve Mamba-3's predictions. To test this hypothesis, we form one block for each documented component or process group and randomly arrange the blocks for each training run. The within-group order is fixed. Thus, the policy specifies which variables stay together while allowing the sequence of groups to vary. Figure 1 shows the nuclear system, its signals, and this ordering operation.

Existing studies already establish that variable order matters in Mamba forecasting. MambaTS learns scan paths from transition costs, SOR-Mamba learns representations that are robust to reversal, and GPS-Mamba constructs graph-based orders (Cai et al., 2026; Lee et al., 2024; Yao et al., 2026). Our question concerns a particular source of ordering information: documented membership of heterogeneous signals in physical components. We compare the ordering policies using the same Mamba-3 architecture (Lahoti et al., 2026) and training conditions.

We study the NPPAD simulation of a nuclear power system (Qi et al., 2022). On held-out simulation records, the grouped policy has lower average MSE than unrestricted random ordering and a GPS-derived history-similarity order. The result supports the grouping hypothesis, but raises a second question: which computations could make grouped signals useful together? Mamba-3's state update identifies two candidates affected by order: accumulated state transport and an additional write from the immediately preceding variable.

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.

We investigate this question in frozen models trained with a fixed component order and matched random orders. First, we test joint use of histories at the output. The same tank-pressure history can affect a prediction differently depending on the other signals; we estimate this dependence and remove it. Next, we test the internal contribution associated with the immediately preceding variable, comparing its removal with a change matched in magnitude and normalization. The first test reverses the fixed order's advantage on the diagnostic windows. The second does not identify the preceding write as its specific source.

Our contribution has two parts. We show that component-grouped orders improve mean test prediction over random and history-similarity orders in this large nuclear system. In the Fixed-grouped–Random diagnostic, removing the estimated output interaction reverses Fixed grouped's advantage on the same validation windows. A controlled internal test does not support the immediately preceding variable's write as the specific explanation of that advantage.

2 From a Nuclear System to Variable Groups

2.1 The physical system and its signals

NPPAD records simulated nuclear-plant operation under accident conditions (Qi et al., 2022). We use 96 channels spanning the reactor cooling system, pressure-control tank, steam generators, injection systems, containment, and associated processes. These include physical quantities with different meanings and units. The task is to predict their future values jointly from their recent histories.

Figure 1 links the plant's components to the variables that Mamba-3 reads. The pressure-control tank, for example, is described by level, enthalpy, and flow signals. We use their common component membership to place them next to one another in the input sequence. We obtain group membership from the source variable descriptions and equipment specifications. Groups include individual devices, subsystem interfaces, and process regions; Appendix A.2 gives the complete mapping used in the experiment.

2.2 Three ways to order the same inputs

Grouped ordering keeps the nonconstant signals of a component in a contiguous block. Let $B_1, \ldots, B_K$ denote these ordered blocks and let ρ_s be a random permutation of the block indices for training seed s. The input order is

$$\pi_s = \mathrm{concat}\big(B_{\rho_s(1)}, \ldots, B_{\rho_s(K)}, U\big), \tag{1}$$

where U contains the training-constant inputs in their fixed trailing order. The 88 nonconstant variables form 18 movable blocks. We draw one distinct block permutation for each training seed and keep it fixed during training and evaluation. This preserves neighbors within each block while varying which blocks meet at their boundaries.

Random ordering draws a distinct permutation of all 96 input variables for each training seed. It imposes no component constraint. GPS-derived ordering uses a graph of similarities between normalized training histories and applies GPS-Mamba's Laplace Scanning (Yao et al., 2026). This gives one fixed order, trained with the same seeds. Appendix B.1 specifies the similarity graph, bandwidth, eigenproblem, and tie handling.

Every policy supplies the same histories and variable identities to the predictor. Histories and identity embeddings move together, and the inverse permutation restores the outputs to their original variable positions. Grouped and Random vary the order across runs; GPS-derived varies training randomness around its fixed order.

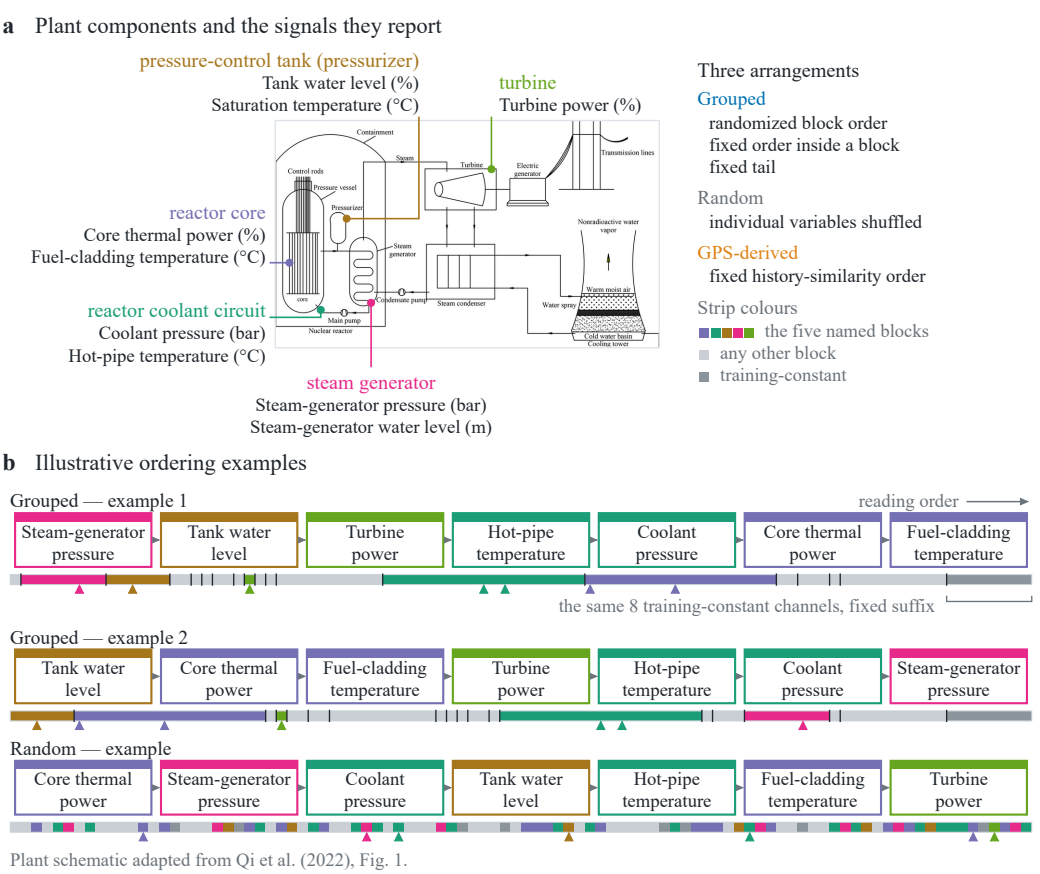

Figure 1: NPPAD: from the plant to variable orders. The plant view connects components to their measured quantities. Grouped keeps a component's signals together while changing the sequence of groups; Random shuffles individual signals. Panel (b) shows illustrative orders. The schematic is adapted from Qi et al. (2022), Fig. 1; source credit and complete membership are given in Appendix A.2.

3 How Order Enters Mamba-3

3.1 A common variable-history predictor

A shared linear projection maps each variable's history to a token, to which we add a learned variable-ID embedding. Input LayerNorm (Ba et al., 2016) is followed by a Mamba-3 SISO core (Lahoti et al., 2026). The same core processes the forward and reversed variable sequences with shared weights. We align and average the two outputs, then apply output LayerNorm and a shared prediction head. Figure 2 separates this predictor from the state update inside the core. Model dimensions and forecasting settings are given in Appendix A.1.

The shared bidirectional construction makes a complete reversal of the input order equivalent after output restoration. Arbitrary permutations still change the neighbors and intermediate variables encountered by each scan. This is the distinction relevant to the Grouped and Random policies.

3.2 Neighbors and state-transport paths

We use the exponential–trapezoidal recurrence and its source-wise expansion from Mamba-3 (Lahoti et al., 2026). In the rotating coordinate frame of one head, the state update is

$$M_i = \alpha_i M_{i-1} + \alpha_i m_i v_{i-1} k_{i-1}^{\mathsf{T}} + \gamma_i v_i k_i^{\mathsf{T}}. \tag{2}$$

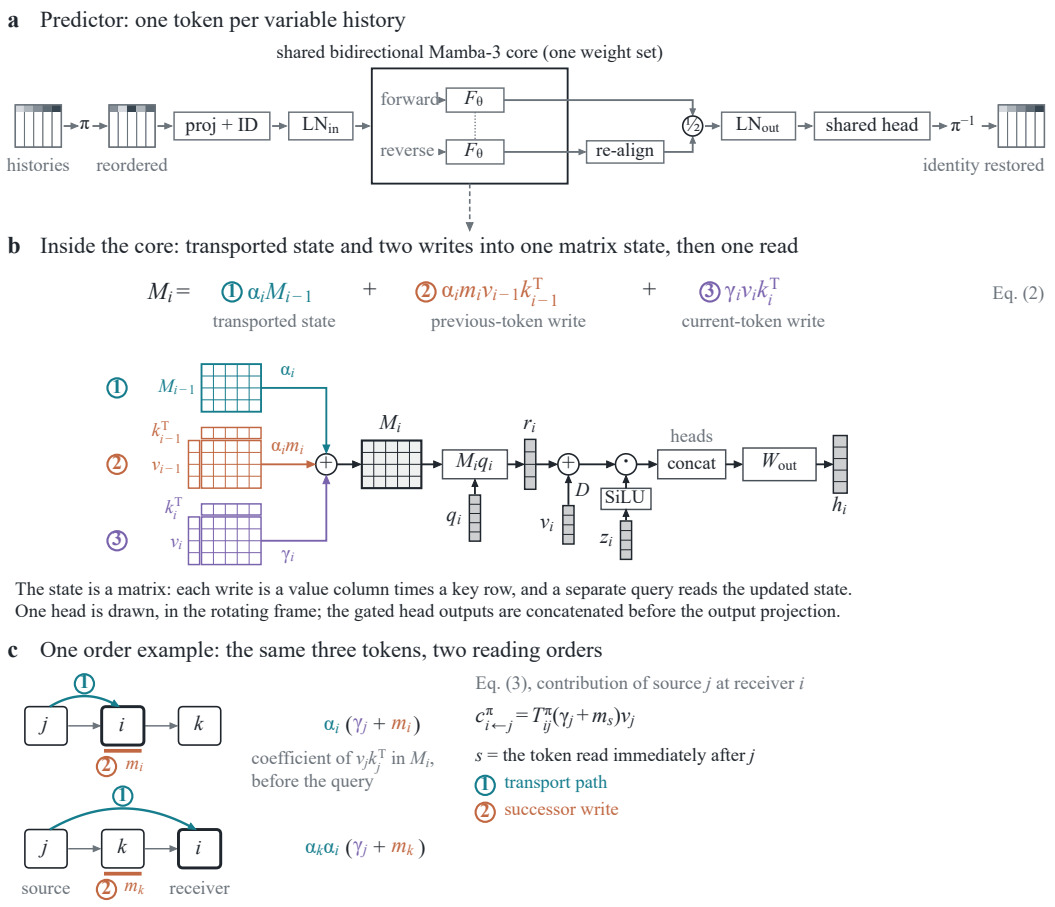

Figure 2: Variable order and the Mamba-3 predictor. Variable histories and identities move together, and the output is restored to the original variable positions. The shared bidirectional core uses the Mamba-3 recurrence: transported state, preceding-token write, and current-token write in Eq. (2). Reordering changes the source's successor and the path to a receiver, as expressed in Eq. (3).

Here i is a position in the variable sequence. The matrix M_i stores accumulated information. The first term transports the preceding state. The second writes the preceding token's value–key outer product once more, and the third writes the current token. A query reads the updated state before the skip connection, gate, and output projection.

In one scan, call an earlier variable j the source and a later variable i the receiver. We examine the source's contribution when the state is read at the receiver. Applying the source-wise expansion to an order π gives

$$c^{\pi}_{i\leftarrow j} = T^{\pi}_{ij}\big(\gamma_j + m_{s_\pi(j\to i)}\big)v_j. \tag{3}$$

The factor T^{π}_{ij} contains the decay, rotation, and query–key contraction along the path. The index $s_\pi(j \to i)$ identifies the token immediately following j in that scan. Changing the order from $j \to i \to k$ to $j \to k \to i$ changes that successor from i to k and puts an additional variable on the path to i. Appendix C derives Eq. (3) from the Mamba-3 recurrence.

Grouped ordering preserves many of these local neighbor relations within physical groups. Across its 18 movable blocks, it preserves 70 within-block adjacent pairs among the 88 nonconstant signals. Random ordering can separate those pairs.

4 TEST-SET COMPARISON

4.1 EVALUATION DESIGN

We divide complete simulation records into training, validation, and test sets. The test records contain different severity conditions within accident families represented in training. A forecast window never crosses a record boundary. We select each model by validation MSE, freeze its checkpoint, and evaluate it on the same 12 test records. Appendix A.1 lists split sizes and preprocessing.

Each ordering policy is trained with 100 seeds. Models with the same seed share their initial weights and minibatch sequence; only their input order differs. Matching these sources of randomness supports a more precise comparison of ordering policies (Bouthillier et al., 2021). The architecture, optimizer, training budget, and checkpoint-selection rule are identical across policies.

The reported score is test MSE in training-standardized units, averaged over windows, forecast leads, and the 88 variables that are nonconstant in training. We compare Grouped with each of Random and GPS-derived using paired run-level differences. The intervals summarize variation across training seeds and, for Grouped and Random, sampled orders. Appendix B.2 gives the paired tests, adjustment across the two comparisons, and paired-bootstrap check.

4.2 GROUPED ORDERS GIVE THE LOWEST MEAN TEST ERROR

Table 1: Prediction on the Nuclear test set. All policies use the same Mamba-3 predictor. Values are the mean and sample standard deviation of test MSE across 100 training seeds. Standard deviation describes run-to-run variation.

Ordering policy	Mean test MSE	Standard deviation
Grouped	**0.055048**	0.002095
Random	0.056270	0.002072
GPS-derived	0.055850	0.002013

Grouped has the lowest mean test MSE (Table 1). Relative to Random, the mean error is 2.17% lower. The paired difference is -1.223×10^{-3} MSE, with a 95% interval of $[-1.741, -0.704] \times 10^{-3}$. Grouped has lower error in 65 of the 100 matched runs.

Relative to GPS-derived ordering, Grouped reduces mean error by 1.44%. The paired difference is -0.802×10^{-3}, with a 95% interval of $[-1.273, -0.332] \times 10^{-3}$. Grouped has lower error in 70 matched runs. The two Holm-adjusted p-values are 1.81×10^{-5} and 0.00102, respectively (Holm, 1979). The paired-bootstrap intervals are negative for both comparisons (Appendix B.2).

The Grouped result averages over distinct permutations of the component blocks. It therefore measures an ordering policy with changing group sequences, as well as changing training seeds. Membership within the blocks is the common structural choice across these runs. The GPS-derived comparison asks whether this documented membership or similarity between training histories gives the lower error when used to order the same model.

4.3 VARIATION ACROSS SIMULATION RECORDS

The aggregate score weights each forecast window equally, so longer records contribute more windows. Grouped has lower mean error than Random in nine records and lower mean error than GPS-derived in nine records. The direction varies across accident conditions. The supplement includes the per-record sufficient statistics.

5 PROBING THE COMPUTATION BEHIND THE ORDERING DIFFERENCE

The prediction comparison tells us which ordering policy works better. We now ask how a model uses the histories it reads together. A natural explanation is that grouping helps it use the target's history jointly with other signals. If this joint dependence contributes to the advantage, removing it should reduce the error gap.

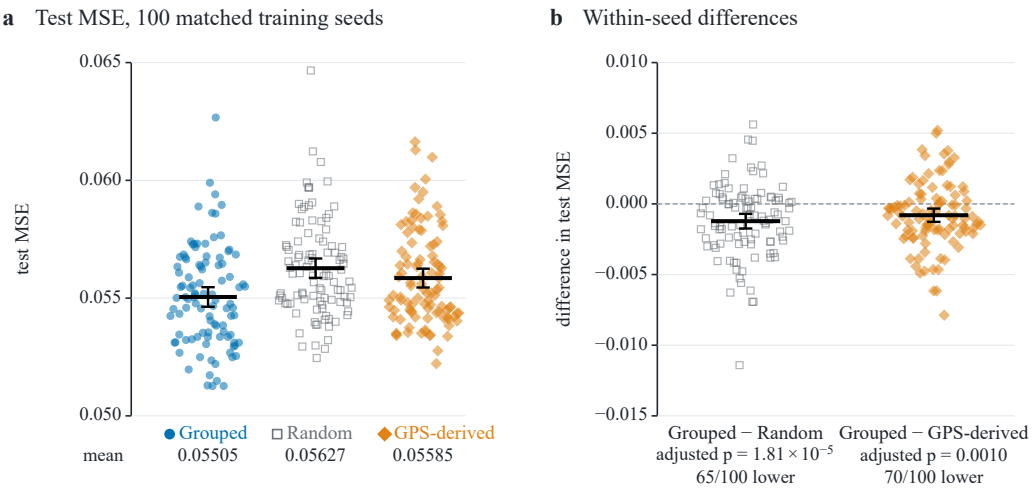

Figure 3: Nuclear test errors and paired ordering differences. Each point represents one training seed; Grouped and Random also use a distinct order per seed. The difference panels subtract the comparator's MSE from Grouped MSE, so negative values favor Grouped. Summary marks show means and pointwise 95% t intervals. P-values are Holm-adjusted over these two comparisons; all policies use the same test records and 88 targets.

We study frozen models trained with one fixed sequence of the component blocks in Section 2, called Fixed grouped, and matched Random models. This diagnostic comparison uses 100 paired seeds and 4,096 shared validation windows per seed, with the same 88 evaluation targets. It examines a fixed order in detail; Section 4 evaluates the randomized Grouped policy on test records. Figure 4 shows the output experiment, followed by a test of the preceding-source path in Figure 2.

5.1 Does joint use of histories contribute to the gap?

A model can use the target history and other histories by adding their separate effects. Here we test whether the effect of the target history changes with the other histories. We call this dependence a non-additive output interaction. This functional definition makes the object of removal explicit (Lengerich et al., 2020; Covert et al., 2021).

Let x be the target history, z the remaining joint histories, and $f(x, z)$ the trained model's prediction for that target. We use a training-reference pair (X'_b, Z'_b) to isolate the dependence:

$$J_b = f(x, z) - f(x, Z'_b) - f(X'_b, z) + f(X'_b, Z'_b). \tag{4}$$

For an additive predictor $f(x, z) = a(x) + b(z)$, the terms cancel. In the experiment, we draw one reference pair from each of two training-reference sets and subtract their mean, $\bar{J} = \frac{J_1 + J_2}{2}$, from the prediction. The remaining histories in each reference come from a single window. No model is retrained. We define the removal cost as the change in MSE after subtracting the estimated interaction:

$$\bar{D} = \mathrm{MSE}(f - \bar{J}) - \mathrm{MSE}(f). \tag{5}$$

We compare the error gap on exactly the same windows before and after removal. Write $g = \mathrm{MSE}_R - \mathrm{MSE}_F$, where F denotes Fixed grouped and R denotes Random. Positive g favors Fixed grouped. After removal,

$$g^- = g - (\bar{D}_F - \bar{D}_R). \tag{6}$$

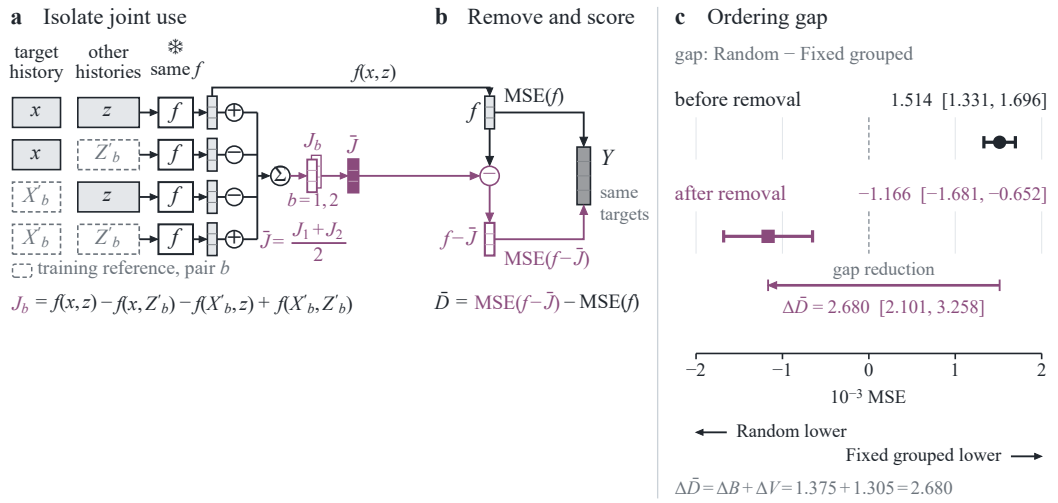

Figure 4: From joint-history dependence to an error-gap test. (a) Four evaluations of the same frozen predictor isolate the reference-based output interaction. (b) The mean of two estimates is subtracted from the prediction. (c) The Random-minus-Fixed-grouped MSE gap changes from positive to negative on the same diagnostic windows. Marks show paired-seed means and pointwise 95% intervals. Fixed grouped uses one component order. Here we compare it with Random on validation windows; Figure 3 reports test performance for randomized Grouped orders. Appendix D specifies sampling and the decomposition of removal cost.

The gap changes from 1.514×10^{-3} to -1.166×10^{-3} MSE (Figure 4). Its reduction is 2.680×10^{-3}, with a 95% interval of $[2.101, 3.258] \times 10^{-3}$. Thus, removing the estimated interaction raises error more for Fixed grouped and reverses its advantage on these windows.

The two reference estimates need not agree. Their disagreement contributes to the measured removal cost through the exact identity $\bar{D} = B + V$, where $B = \langle Y - f, J_1 + J_2 \rangle + \langle J_1, J_2 \rangle$ and $V = \frac{\|J_1 - J_2\|^2}{4}$. The Fixed-grouped-minus-Random differences are 1.375×10^{-3} for B and 1.305×10^{-3} for V. Reference disagreement therefore contributes to the observed sign change. Appendix D.2 reports them separately, including the adjusted tests. We next ask whether a particular internal write accounts for the difference.

5.2 Is the preceding variable's write the explanation?

Equation (2) gives the immediately preceding variable a direct path into the current state. We isolate its reference-centered contribution, including both its write retained in the state and its additional write at the receiver. After aligning the two scan directions, we remove this term before output LayerNorm and apply the same trained prediction head.

Changing an internal vector can raise error because the prediction changes. We therefore compare the preceding-source removal with a change in another direction, matched in magnitude and in the LayerNorm denominator. For each intervention, we subtract the removal cost for Random from that for Fixed grouped. If the preceding source specifically favors Fixed grouped, the resulting difference should be larger for preceding-source removal than for the matched control.

The measured difference is -1.438×10^{-3} MSE, with a 95% interval of $[-2.140, -0.736] \times 10^{-3}$. The control produces the larger excess cost. This result does not support the immediately preceding source as a specific explanation of the fixed order's advantage. The output test identifies a difference in estimated joint-history dependence, but the internal test does not identify its specific source. Appendix D.3 defines the removed term and the matched control.

6 Related Work

Multivariate forecasters differ in how they share information between variables. PatchTST applies shared channel-independent processing (Nie et al., 2023), whereas iTransformer represents each variable as a token and uses attention across variables (Liu et al., 2024). CCM shares information through learned channel clusters based on similarity (Chen et al., 2024). For physical measurements, documented component membership supplies another grouping criterion: signals may describe the same equipment while measuring different quantities. Our experiment studies that criterion as an input-order prior.

State-space forecasting methods provide several ways to combine temporal and cross-variable information. S-Mamba scans the variable axis bidirectionally (Wang et al., 2024), while Chimera models temporal and variate dimensions with a two-dimensional state-space design (Behrouz et al., 2024). MambaTS learns scanning choices from variable-transition costs (Cai et al., 2026). SOR-Mamba seeks robustness to sequential order through reversed-input consistency (Lee et al., 2024). We hold the forecasting architecture fixed and compare the information used to arrange its variable tokens.

GPS-Mamba is the closest graph-based ordering approach (Yao et al., 2026). It uses distance or history-similarity graphs to construct a Laplace Scanning order, together with additional forecasting branches. We apply its history-similarity ordering procedure to the same Mamba-3 predictor used by the other policies. This compares similarity-derived order with a component-grouped policy whose block sequence varies across runs. The graph and ordering implementation are specified in Appendix B.1.

Mamba-3 improves state-space sequence modeling through its discretization and state representation (Lahoti et al., 2026), building on selective state-space models and state-space duality (Gu & Dao, 2024; Dao & Gu, 2024). Studies of input selectivity and token dynamics examine how these models retain and transform information (Huang et al., 2025; Vo et al., 2025). Functional interaction analysis distinguishes joint dependence from additive effects (Lengerich et al., 2020), and removal-based explanations measure how changing an identified part of a predictor affects its output (Covert et al., 2021). We use the Mamba-3 recurrence to define a candidate internal path, then examine output interaction and that path under matched comparisons.

7 Discussion and Conclusion

The Nuclear test comparison supports a practical design choice: keep a component's heterogeneous signals together when constructing the variable sequence. Grouped orders improve average prediction over unrestricted permutations and history-similarity ordering under the same model and training conditions. Their block sequence varies between runs, so the policy leaves flexibility in the global sequence while preserving local groups.

The computational experiments address the next question: how could an order help? In the fixed-order diagnostic, removing the estimated target–rest interaction reverses the error advantage. The preceding-source term, however, has no larger order-specific removal cost than the magnitude- and normalization-matched control. These results favor investigating how information from multiple sources is accumulated and read, rather than assigning the effect to the nearest preceding variable alone.

7.1 Limitations

This is an exploratory study of one simulated nuclear system. Its test records vary accident severity within known accident families. The intervals quantify training and ordering variation on a fixed test set. Grouped fixes within-component order and places training-constant inputs at the end; the Random comparison includes these choices. The computational probes use a fixed component order on validation windows and two reference estimates per window. Extending them to shuffled group orders and larger reference samples would test how far the explanation transfers. Public datasets combining many variables, heterogeneous quantities, and documented components remain limited.

7.2 Conclusion

Which variables Mamba-3 reads together affects prediction in this large nuclear-system task. Component-grouped orders lower mean test error. On the diagnostic windows, removing the estimated interaction between a target history and the other histories reverses Fixed grouped's advantage over Random. The matched internal control does not support the preceding variable's write as a specific explanation of that advantage.

AI use statement

The authors established the research direction and led the drafting of the manuscript. OpenAI and Anthropic tools assisted with manuscript revision and English-language editing, as well as literature search, experiment design, mathematical checks, coding, data analysis, and figure preparation. Reported statistics were recomputed from saved sufficient statistics, and references were checked against source publications. The authors take responsibility for the study and final manuscript.

Ethics statement

We use publicly available simulated data and identify no additional ethical concerns arising from this study.

Reproducibility statement

Appendix A specifies the data, component mapping, and training configuration; Appendix B gives the graph-based comparator and statistics; Appendix C connects the variable order to the Mamba-3 recurrence; Appendix D defines the computational probes. The anonymous supplement includes input orders, model and training code, evaluation configuration, sufficient statistics, and scripts for recomputing the reported results. Raw simulation data are obtained from the cited provider.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450v1.

Ali Behrouz, Michele Santacatterina, and Ramin Zabih. Chimera: Effectively modeling multivariate time series with 2-dimensional state space models. In *Advances in Neural Information Processing Systems 37*, 2024. URL https://papers.nips.cc/paper_files/paper/2024/hash/d8e80772c27beff4ae1676fb147bbf26-Abstract-Conference.html.

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Naz Sepah, Edward Raff, Kanika Madan, Vikram Voleti, Samira Ebrahimi Kahou, Vincent Michalski, Dmitriy Serdyuk, Tal Arbel, Chris Pal, Gael Varoquaux, and Pascal Vincent. Accounting for variance in machine learning benchmarks. In *Proceedings of Machine Learning and Systems*, volume 3, pp. 747–769, 2021. URL https://proceedings.mlsys.org/paper_files/paper/2021/file/0184b0cd3cfb185989f858a1d9f5c1eb-Paper.pdf.

Xiuding Cai, Xueyao Wang, Yaoyao Zhu, and Yu Yao. MambaTS: Improved selective state space models for long-term time series forecasting. *Pattern Recognition*, 180:114536, 2026. doi: 10.1016/j.patcog.2026.114536. URL https://doi.org/10.1016/j.patcog.2026.114536.

Jialin Chen, Jan Eric Lenssen, Aosong Feng, Weihua Hu, Matthias Fey, Leandros Tassiulas, Jure Leskovec, and Rex Ying. From similarity to superiority: Channel clustering for time series forecasting. In *Advances in Neural Information Processing Systems 37*, 2024. URL https://papers.nips.cc/paper_files/paper/2024/hash/eb9b18ccb76a1156af5779ffdca1d91f-Abstract-Conference.html.

Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021. URL https://www.jmlr.org/papers/v22/20-1316.html.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10041–10071. PMLR, 2024. URL https://proceedings.mlr.press/v235/dao24a.html.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7 (1):1–26, 1979. doi: 10.1214/aos/1176344552. URL https://doi.org/10.1214/aos/1176344552.

Albert Gu and Tri Dao. Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling*, 2024. URL https://openreview.net/pdf?id=tEYskw1VY2.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979. URL https://www.jstor.org/stable/4615733.

Ningyuan Teresa Huang, Miguel Sarabia, Abhinav Moudgil, Pau Rodriguez, Luca Zappella, and Federico Danieli. Understanding input selectivity in Mamba: Impact on approximation power, memorization, and associative recall capacity. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 25693–25727. PMLR, 2025. URL https://proceedings.mlr.press/v267/huang25ab.html.

Aakash Sunil Lahoti, Kevin Li, Berlin Chen, Caitlin Wang, Aviv Bick, Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/8abd2043b71a074278d5f687947bff9c-Abstract-Conference.html.

Seunghan Lee, Juri Hong, Kibok Lee, and Taeyoung Park. Sequential order-robust Mamba for time series forecasting, 2024. URL https://arxiv.org/abs/2410.23356v1.

Benjamin Lengerich, Sarah Tan, Chun-Hao Chang, Giles Hooker, and Rich Caruana. Purifying interaction effects with the functional ANOVA: An efficient algorithm for recovering identifiable additive models. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2402–2412. PMLR, 2020. URL https://proceedings.mlr.press/v108/lengerich20a.html.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/2ea18fdc667e0ef2ad82b2b4d65147ad-Abstract-Conference.html.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101v3.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2211.14730v2.

Ben Qi, Xingyu Xiao, Jingang Liang, Li-chi Cliff Po, Liguo Zhang, and Jiejuan Tong. An open time-series simulated dataset covering various accidents for nuclear power plants. *Scientific Data*, 9:766, 2022. doi: 10.1038/s41597-022-01879-1. URL https://doi.org/10.1038/s41597-022-01879-1.

Thieu N. Vo, Tung D. Pham, Xin T. Tong, and Tan Minh Nguyen. Demystifying the token dynamics of deep selective state space models, 2025. URL https://arxiv.org/abs/2410.03292v2.

Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang, and Yifei Zhang. Is Mamba effective for time series forecasting?, 2024. URL https://arxiv.org/abs/2403.11144v3.

Zihao Yao, Qi Zheng, Jiankai Zuo, and Yaying Zhang. GPS-Mamba: Graph permutation scanning state space model for multivariate time series forecasting. *Expert Systems with Applications*, 311: 131373, 2026. doi: 10.1016/j.eswa.2026.131373. URL https://doi.org/10.1016/j.eswa.2026.131373.

A Data, Groups, and Training

A.1 Dataset and prediction settings

We use the NPPAD CSV records from the dataset authors' public repository, revision a8d54ed0 (Qi et al., 2022). The source observations are separated by 10 seconds. No temporal aggregation is applied. Each window contains 24 input observations and 12 future observations and advances by one row within a record.

The split contains 65 training, 12 validation, and 12 test records. It separates complete severity-specific simulations within accident families. Families represented by a single record remain in training. A duplicate simulation record was excluded before window construction. One 11-row training record contributes to normalization but is too short to yield a forecast window. The supplement supplies the exact record lists and hashes.

For input variable i, training observations determine mean μ_i and standard deviation s_i. We transform observations as

$$z_{ti} = \frac{x_{ti} - \mu_i}{\max(s_i, 10^{-6}|\mu_i|, 10^{-6})}. \tag{A1}$$

The model takes and predicts all 96 channels. Training-constant input channels are set to zero after standardization. Checkpoint selection and reported test MSE use the same 88 training-nonconstant channels. Normalization is fixed from training data.

Table A1: Data and model settings. All ordering policies share these settings.

Setting	Value
Input / output channels	96 / 96
Evaluation targets	88 training-nonconstant channels
Training / validation / test records	65 / 12 / 12
Training / validation / test windows	25,797 / 4,928 / 4,876
History / forecast horizon	24 / 12 observations
Sampling interval / window stride	10 seconds / 1 observation
Token / state dimension	64 / 32
Core expansion / heads / groups	2 / 4 / 1
Parameters	38,996
Training epochs / optimizer updates	80 / 8,080
Batch size	256
Optimizer / learning rate	AdamW / 0.001
AdamW betas / epsilon / weight decay	(0.9, 0.999) / 10^{-8} / 0.01
Gradient clipping	Global norm 1

We train with AdamW (Loshchilov & Hutter, 2019) to minimize MSE across all output channels and forecast leads. There is no early stopping or learning-rate schedule. We retain the first epoch checkpoint attaining the minimum validation MSE on the 88 evaluation targets, then evaluate that frozen model on the test records. The native core uses BF16 activations; selected factors and the adapter readout use FP32. Test squared-error sums are accumulated in FP64.

Let N be the number of forecast windows, H the horizon, and $\mathcal{T}$ the evaluation targets. In the standardized coordinates of Eq. (A1),

$$\mathrm{MSE} = \frac{1}{NH|\mathcal{T}|} \sum_{n=1}^{N} \sum_{h=1}^{H} \sum_{i \in \mathcal{T}} (\widehat{z}_{nhi} - z_{nhi})^2. \tag{A2}$$

The test denominator is $4{,}876 \times 12 \times 88 = 5{,}149{,}056$. This is a window-weighted mean; each simulation record contributes in proportion to its window count. MSE units and target membership are identical across policies.

A.2 Component membership and complete input orders

The source system schematic and simulator descriptions identify equipment, interfaces, and process regions (Qi et al., 2022). Table A2 lists the 18 blocks containing the 88 nonconstant variables. Each block retains the channel order in the supplied mapping. Grouped permutes the blocks using a separate layout seed and appends the eight remaining input channels. The supplement contains all 100 realized orders and the exact generation code.

The plant schematic in Figure 1 is adapted from Qi et al. (2022), Fig. 1, under CC BY 4.0, with added annotations and ordering examples.

Table A2: Movable component/process blocks. Counts refer to nonconstant input variables. These blocks contain 88 variables in total.

Component or process	Variables
Auxiliary building	1
Exclusion-area boundary receptor	2
Steam-generator tube interface A	1
Steam-generator tube interface B	1
Chemical/volume-control and coolant interface	2
Turbine	1
Steam and feedwater	10
Steam generators	8
Pressure-control tank (pressurizer)	6
Primary reactor coolant system	19
Emergency core cooling	3
Condenser and off-gas	2
Reactor core	18
Lower plenum	1
Refueling-water storage tank	1
Containment	10
Reactor cavity	1
Break-path signals	1

The source diagram locates the components it depicts; process regions and receptor signals are identified from the variable descriptions. The complete map includes names, source columns, units, and group membership. The eight training-constant inputs are excluded from the reported targets and remain at the end of every Grouped order. Random permutes all 96 channels; GPS-derived orders all 96 using the graph in Appendix B.1.

B Comparator and Statistical Details

B.1 GPS-derived history-similarity order

We apply the history-similarity graph and Laplace Scanning procedure from GPS-Mamba (Yao et al., 2026, Eqs. 15 and 20) to the input order of the shared Mamba-3 predictor. The comparison uses that ordering module; the forecasting architecture remains the model in Section 3.

For each variable i, vector x_i concatenates normalized training input windows in window and within-window time order. Thus overlapping observations occur repeatedly, as they do in the constructed training windows. Let $d_{ij}^2 = \|x_i - x_j\|_2^2$. The graph and ordering solve

$$W_{ij} = \exp\left(-\frac{d_{ij}^2}{2\sigma^2}\right), \qquad D_{ii} = \sum_j W_{ij}, \qquad L = D - W, \qquad Lz = \lambda Dz. \tag{B1}$$

We set σ^2 to half the median of positive off-diagonal squared distances and retain $W_{ii} = 1$. The order sorts coordinates of the eigenvector associated with the smallest positive eigenvalue. Adjacent coordinates u, v satisfying $|u - v| \le 10^{-12} + 10^{-10}\max(1, |u|, |v|)$ form a tie block, with column indices breaking ties. We select the lexicographically smaller of the two block directions. The saved order agrees across normalized and generalized eigenproblem solvers. The graph and order use training histories only and remain fixed for all 100 GPS-derived runs.

B.2 Paired comparisons

For comparator O, define the seed-level difference $d_s = \mathrm{MSE}_{\mathrm{Grouped},s} - \mathrm{MSE}_{O,s}$. Let $\bar{d}$ and s_d be its sample mean and standard deviation over $n = 100$ paired seeds. We use the two-sided paired t-test and pointwise interval

$$t = \frac{\bar{d}}{\frac{s_d}{\sqrt{n}}}, \qquad p = 2\{1 - F_{t_{n-1}}(|t|)\}, \qquad \mathrm{CI}_{95\%} = \bar{d} \pm t_{n-1,.975}\frac{s_d}{\sqrt{n}}. \tag{B2}$$

Holm adjustment is applied jointly to the Grouped–Random and Grouped–GPS-derived comparisons (Holm, 1979). The paired-seed bootstrap uses 20,000 resamples, retaining all policies of a seed together, and reports the 2.5th and 97.5th percentiles (Efron, 1979). The analysis uses complete 100-seed cohorts.

Table B1: Grouped minus comparator test MSE. Effects and pointwise 95% t intervals are in 10^{-3} MSE. Negative values favor Grouped. P-values are adjusted across the two rows.

Comparator	Difference	95% interval	Holm-adjusted p	Grouped lower
Random	-1.223	$[-1.741, -0.704]$	1.81×10^{-5}	65/100
GPS-derived	-0.802	$[-1.273, -0.332]$	0.00102	70/100

The supplement includes the full paired values needed to recompute means, differences, intervals, and tests.

C Order in the Mamba-3 Recurrence

Equation (2) and the source-wise expansion follow Mamba-3 (Lahoti et al., 2026, Sec. 3.2.2 and Appendices B.3.1 and C.3). In one scan and head, $\alpha_i = \exp(A_i\Delta_i)$, $m_i = (1 - \sigma(\tau_i))\Delta_i$, and $\gamma_i = \sigma(\tau_i)\Delta_i$, with σ denoting the sigmoid function. For a source at position $j < i$, its write enters at j and once more at $j + 1$. Expanding the recurrence gives its coefficient at receiver i:

$$\gamma_j \prod_{t=j+1}^{i} \alpha_t + \alpha_{j+1} m_{j+1} \prod_{t=j+2}^{i} \alpha_t = (\gamma_j + m_{j+1}) \prod_{t=j+1}^{i} \alpha_t. \tag{C1}$$

Combining the transport product with rotation and the query–key contraction yields T_{ij}^{π} in Eq. (3). On the diagonal, only the current write appears. At the first token, the previous-token write is zero. The successor in the reverse scan is defined in that scan's direction.

The two directions share parameters. For a token matrix Z, their aligned output is

$$F_{\mathrm{bi}}(Z) = \frac{1}{2}\left\{F_\theta(Z) + \mathrm{Rev}\, F_\theta(\mathrm{Rev}\, Z)\right\}. \tag{C2}$$

Reversing the complete token sequence exchanges these two terms. After restoring variable positions, the prediction is unchanged. A completely reversed sequence therefore implements the same ordering policy in this predictor. Moving individual variables, however, changes the source's successor and the intermediate transport path in Eq. (3).

D Output Interaction and Internal Removal

D.1 Frozen models and reference sampling

The diagnostic uses 100 Fixed-grouped models and 100 Random models, paired by training seed. Fixed grouped uses one fixed concatenation of the same 18 blocks and trailing inputs defined in Section 2. The supplement supplies this order separately from the randomized Grouped orders in Section 4. Models are frozen at their validation-selected checkpoints. For each seed, 4,096 of the 4,928 validation windows are sampled and shared by both arms. Scores average the 88 nonconstant targets and 12 forecast leads using the scales in Appendix A.1.

For each seed, two nonoverlapping sets of 256 training windows are sampled without replacement. For each evaluation window and each reference set, two indices are drawn independently and uniformly. The target history is taken from the first indexed window and all remaining histories from the second. These indices are shared by all targets and both models at that seed. Thus there are two interaction estimates per evaluation window, one from each reference set. Target values do not select references. Model, order, evaluation-window, and reference identities are included in the supplement.

The four evaluations in Eq. (4) define an interaction relative to a reference pair. Under a product reference distribution for target and remaining histories, its expectation is

$$\mathcal{I}(x,z) = f(x,z) - \mathbb{E}_{Z'} f(x,Z') - \mathbb{E}_{X'} f(X',z) + \mathbb{E}_{X',Z'} f(X',Z'). \tag{D1}$$

This is zero for an additive predictor. In the experiment, Eq. (5) removes the mean of the two reference estimates, $\bar{J}$. We aggregate windows, targets, and leads within each model before taking differences across the 100 paired seeds.

D.2 Removal cost and reference disagreement

Let $r = Y - f(x,z)$ and let inner products include the averaging used by MSE. Expanding Eq. (5) gives

$$\begin{aligned} B &= \langle r, J_1 + J_2\rangle + \langle J_1, J_2\rangle, \\ V &= \tfrac{1}{4}\|J_1 - J_2\|^2, \qquad \bar{D} = B + V. \end{aligned} \tag{D2}$$

B is the two-reference statistic specified for the diagnostic; sampled-mean removal cost was reconstructed subsequently. The V term records disagreement between the reference estimates. For fixed reference sets, put $\mu_b = \mathbb{E}[J_b]$. Conditional independence of the two draws gives

$$\mathbb{E}[B] = \mathrm{MSE}\big(f - \tfrac{\mu_1+\mu_2}{2}\big) - \mathrm{MSE}(f) - \tfrac{1}{4}\|\mu_1 - \mu_2\|^2. \tag{D3}$$

Thus B and actual sampled-mean removal cost are reported separately. The mean baseline MSE on these diagnostic windows is 0.093172 for Fixed grouped and 0.094685 for Random. Mean removal costs are 0.007383 and 0.004704, respectively. The after-removal gap is computed within each paired seed before averaging.

Table D1: Nuclear computational probes. Differences are Fixed grouped minus Random, except the before/after gaps, which are Random minus Fixed grouped. All effects and pointwise 95% intervals are in 10^{-3} MSE. The internal contrast T is defined in D.3. Adjusted p-values retain the complete diagnostic-study families; all family members and raw values are supplied in the supplement.

Measure	Mean	95% interval	Holm-adjusted p
Gap before removal	1.514	[1.331, 1.696]	—
Gap after removal	-1.166	$[-1.681, -0.652]$	—
Removal-cost difference $\Delta\bar{D}$	2.680	[2.101, 3.258]	1.27×10^{-14}
Two-reference difference ΔB	1.375	[0.958, 1.792]	1.09×10^{-8}
Disagreement difference ΔV	1.305	[1.041, 1.568]	—
Internal matched-control contrast T	-1.438	$[-2.140, -0.736]$	2.91×10^{-4}

D.3 Preceding-source term and matched control

For one scan direction, express the state in the local frame at the preceding position. Let H be the state before the receiver, E the predecessor's value–key outer product, and $G = \gamma_{\text{prev}} E$ its write retained in H. At receiver i, define $(A_0)_{hvk} = \alpha_{ih} g_{ihv} [R(\delta_{ih}) q_{ih}]_k$ and $A_1 = m_i A_0$. The gate g is the native SiLU gate and δ is the rotary increment.

For each model and reference set, $F^c = F - \mathbb{E}_{\text{ref}} F$ centers each tensor over its 256 training windows. We form A_1 and G before centering. Let $\mathcal{P}(F) = \text{concat}_h(\sum_k F_{h,:,k})$ and $O \in \mathbb{R}^{64\times 128}$ be the core output projection. The directional preceding-source term is

$$J_{\text{prev}} = \tfrac{1}{2} O \mathcal{P} \left(A_0^c \odot G^c + A_1^c \odot E^c \right). \tag{D4}$$

The two terms cover that source's retained write and its additional write at the receiver. Directional contributions are restored to variable positions and summed; the factor $\frac{1}{2}$ is applied once. We subtract the combined term before output LayerNorm, keeping the trained head fixed. This reference-centered internal term is defined separately from the output interaction in Eq. (4).

For the pre-LayerNorm representation u and removal vector s, center across features to obtain u_c and s_c. A donor vector v for the same target is selected in advance from another training reference. We project its centered value away from u_c and construct

$$\begin{aligned} p &= \frac{\langle u_c, s_c \rangle}{\|u_c\|^2} u_c, \qquad v_\perp = v_c - \frac{\langle u_c, v_c \rangle}{\|u_c\|^2} u_c, \\ s^{\text{ctrl}} &= \bar{s}\mathbf{1} + p + \|s_c - p\| \frac{v_\perp}{\|v_\perp\|}. \end{aligned} \tag{D5}$$

The control matches the removal mean, centered norm, and inner product with u_c. Consequently $\|u_c - s_c^{\text{ctrl}}\|^2 = \|u_c - s_c\|^2$, giving the same LayerNorm denominator. Its direction differs. Both removed representations pass through the same LayerNorm affine transform and prediction head, which remain sensitive to the removal direction. Donors are chosen without using target values or removal costs; invalid or degenerate matches are recorded without redrawing.

Let $D_F^{\text{prev}}, D_R^{\text{prev}}$ be preceding-source removal costs and $D_F^{\text{ctrl}}, D_R^{\text{ctrl}}$ the matched-control costs. The tested contrast is

$$T = (D_F^{\text{prev}} - D_R^{\text{prev}}) - (D_F^{\text{ctrl}} - D_R^{\text{ctrl}}). \tag{D6}$$

A positive value would indicate a larger order-specific cost for the preceding-source removal. The negative estimate in Table D1 gives the opposite ordering between the two interventions.

D.4 Numerical verification and inference

Unmodified full-validation MSE was reproduced for all 200 diagnostic models. On the current 88-target scores, $|B - A - E|$ is at most 2.26×10^{-17}, where $A = \langle r, J_1 + J_2 \rangle$ and $E = \langle J_1, J_2 \rangle$.

Direct sampled-mean removal cost agrees with Eq. (D2) within 2.43×10^{-17}; the before/after gap identity agrees within 2.78×10^{-17}.

Means and pointwise t intervals use 100 paired seed-level differences. Evaluation windows, targets, leads, and reference sets are averaged within each model. The two-reference statistic and the internal contrast retain their joint multiplicity adjustment; the subsequent sampled-mean removal comparison retains its separate adjustment. The supplement provides the complete inputs for both calculations.