# STATISTICAL STRUCTURE AS AN INTERFACE TO FROZEN COMPUTATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

When a frozen model's parameters cannot be changed and no dedicated control input can be added, statistical structure in its ordinary context can still serve as an interface to its computation. Using Prior-Data Fitted Networks (PFNs) as a controlled testbed, we construct training distributions in which a latent control state is expressed only through a statistical property of the context, making that property a *carrier* of the state. After training, rewriting this carrier alone causally selects which computation the frozen model performs, with graded, multi-state control. How readily such an interface is learned is not fixed by the information it provides: across 48 carriers calibrated so that a Bayes-optimal decoder recovers the state from a 128-row context with the same accuracy, the number of training steps a PFN needs to read the carrier varies twelve-fold with its statistical structure. Interventions on prior mass, evidence strength, and training support show that a carrier's meaning is assigned by the training distribution: within the studied PFN families, predictions closely track the posterior it implies, multiple carriers combine according to learned likelihoods, and state combinations absent from training are increasingly suppressed with continued optimization. In pretrained language models, statistical properties of demonstrations already provide graded, multi-state behavioral control without any deliberate interface training. A property that shifts behavior, however, cannot in general be assigned an arbitrary new meaning by in-context demonstrations, and whether it can depends strongly on how the property is concretely instantiated in the prompt. Together, these results establish statistical structure as a writable input-side interface to frozen computation and characterize the principles governing its acquisition, meaning, and limits.

## 1 INTRODUCTION

Frozen models can contain many computations without exposing an interface for selecting among them. An instruction need not implement a computation; it can select behavior learned during training. The same can hold for statistical structure: if the sign of a correlation in context is associated with one of two functions (Figure 1), changing that sign can switch the model's computation. We call such a property a *carrier* and ask when it becomes a *writable interface*: a state that can be written into context, read by the model, and used to select computation.

Transformers can extract statistical structure from context, including covariance and higher-order dependencies (Xu & Fang, 2026; Gibson et al., 2026), while in-context learning can infer latent tasks, implement learning algorithms, and select computations (Xie et al., 2022; Akyürek et al., 2023; Von Oswald et al., 2023; Bai et al., 2023). Frozen models can also be steered through optimized textual prompts (Büssing et al., 2026; Agrawal et al., 2026; Zehle, 2026), internal representations (Hendel et al., 2023; Todd et al., 2024; Rimsky et al., 2024), and structured inputs (Wayment-Steele et al., 2024; Franke et al., 2026; Gut et al., 2026). What remains largely uncharacterized is which properties of ordinary input become control variables, how readily they are acquired, and what determines their meaning and limits.

Answering this requires separating available information from what the learner actually acquires. Statistically available structure can differ sharply in learning difficulty (Barak et al., 2022; Abbe et al., 2022; 2023), while training distributions shape in-context inference and generalization beyond training support (Chan et al., 2022; Singh et al., 2023; Raventós et al., 2023). Prior-Data Fitted



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

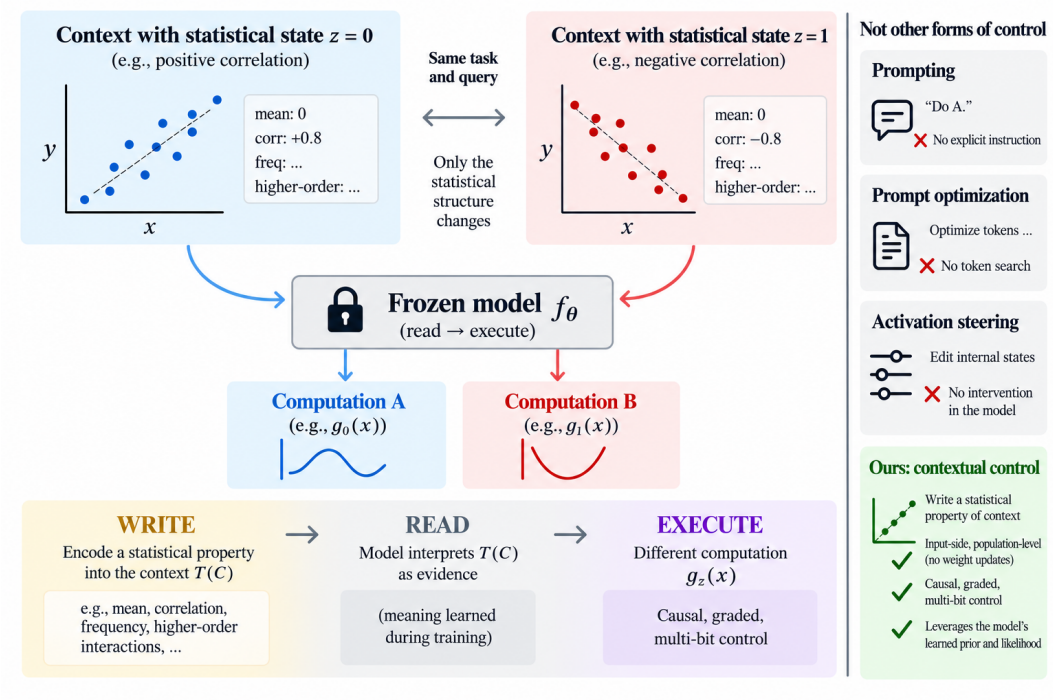



Figure 1: **Statistical structure as an interface to frozen computation.** A latent control state is written into statistical properties of ordinary context; rewriting the carrier after training can change the computation of the frozen model.

Networks (PFNs), transformers trained on datasets sampled from a specified prior to approximate the corresponding posterior predictive, provide a controlled laboratory because their training distribution and inference problem are explicitly specified (Müller et al., 2022; Nagler, 2023; Reuter et al., 2025). We can therefore match carriers for finite-context information, intervene on them after training, and independently manipulate the prior, likelihood, and support that give them meaning.

Using PFNs, we find causal, graded, and multi-state control through statistical context. Across 48 prospectively sampled carriers matched for decodability, acquisition varies strongly with statistical structure, while multiple interfaces can coexist with little cross-talk once acquired. Their semantics are set by the training distribution: predictions track its posterior, carriers combine according to learned likelihoods, and unsupported compositions are increasingly suppressed with optimization. Statistical context can also communicate a latent task prior and select structured computations. In pretrained language models, statistical properties already provide graded, multi-state, and simultaneous behavioral control, but sensitivity does not imply arbitrary semantic bindability, which depends strongly on the carrier's physical realization; computation selection in the stronger sense is established only in the controlled PFN setting. Code and frozen result records are available in the anonymous repository: https://anonymous.4open.science/r/statistical-structure-interface-EA55.

**Contributions.**

- **Writable statistical interfaces.** We establish statistical structure as a causal, writable, multi-state interface to frozen computation and characterize its acquisition and coexistence.
- **Interface semantics and limits.** We show how training-distribution priors, learned likelihoods, and support govern interface semantics, evidence fusion, and compositional boundaries.
- **Interfaces under natural pretraining.** We show that related interfaces arise under natural language pretraining while separating broad statistical sensitivity from selective, realization-dependent bindability.

Extended related work and further positioning are provided in Appendix A.

## 2 STATISTICAL STRUCTURE AS A WRITABLE INTERFACE

A latent state $z \in \{0, \dots, K-1\}$ determines which computation $g_z$ applies to query $x$, but is never given to the model. Instead, a carrier family $\mathcal{T}$ encodes information about $z$ in a statistical property $T(C)$ of context $C$:

$$z \sim p(z), \qquad C \sim p_{\mathcal{T}}(C \mid z), \qquad y = g_z(x). \tag{1}$$

After training, the model is frozen and we intervene only on the carrier. We call an interface *writable* when its state can be written into context, read by the model, and executed as the corresponding computation.

### 2.1 A CONTROLLED STATISTICAL INTERFACE

The Bayes-optimal predictor averages state-dependent computations according to the evidence in context,

$$f^\star(C, x) = \sum_z p(z \mid C)\, g_z(x). \tag{2}$$

When state dependence factors through the carrier, $p(z \mid C) = p(z \mid T(C))$; more generally, we quantify the information available from a finite context using the corresponding ideal observer. This separates information availability from acquisition by the learner.

We distinguish READ, where the model recovers $z$, from EXECUTE, where the inferred state selects a separate computation $g_z$ applied to a query operand. The models are trained separately, so either capability can fail independently. PFNs provide a controlled learner because their synthetic training distribution makes both the available evidence and its meaning explicit (Müller et al., 2022; Nagler, 2023; Reuter et al., 2025). Unless stated otherwise, $N = 128$ and carrier strengths are calibrated to approximately $90\%$ finite-context ideal-observer accuracy. Full constructions and controls are given in Appendix B.

### 2.2 REWRITING STATISTICAL STRUCTURE CONTROLS FROZEN COMPUTATION

Across 12 audit-clean carriers spanning marginal moments, dependencies, missingness, and geometric structure, 10 reach at least $90\%$ of the corresponding ideal-observer span in all three seeds on separately trained READ and EXECUTE models (Figure 2a). Statistical context can therefore do more than identify a latent state: it can select which computation a frozen model performs (Appendix C.1).

To test writability, we construct matched contexts that differ only in carrier state while query, task, base randomness, and frozen model remain fixed. For 9 of these 10 carriers, rewriting the carrier rewrites the selected computation in every seed, reaching at least $90\%$ of the ideal-observer pair rate (Figure 2c).

Control is graded: mean state-following rises from $0.501$ at zero evidence to $0.896$ at training strength and to $0.968$, $0.991$, and $0.997$ under stronger evidence, closely tracking the ideal observer (Figure 2b). Conversely, `anisotropy` can be read near its ideal observer in two of three seeds yet fails to reliably control EXECUTE. *Readability therefore does not guarantee executability.*

A single carrier can also select multiple computations. At $K = 8$, covariance, fourth-order parity, mean, and variance reach $0.92$–$0.98$ of task-oracle performance across all seeds after the specified $30{,}000$-step extension, demonstrating three executable control bits under the tested construction (Figure 2d). Full intervention and capacity results are in Appendices C.2 and C.3.

### 2.3 MATCHED DECODABILITY DOES NOT DETERMINE INTERFACE ACQUISITION

If finite-context decodability alone determined acquisition, carriers matched for ideal-observer accuracy should be learned similarly. We test 48 interfaces sampled prospectively from a structural-equation DSL and calibrated before any PFN outcome is observed. All $144$ runs cross $90\%$ of the above-chance ideal-observer span within the training budget, yet first-crossing times span $250$–$5{,}000$ steps and carrier-level mean acquisition costs vary twelve-fold (Figure 3). Higher-order and nonlinear structure are associated with slower acquisition; five descriptors fixed before training explain $66.4\%$

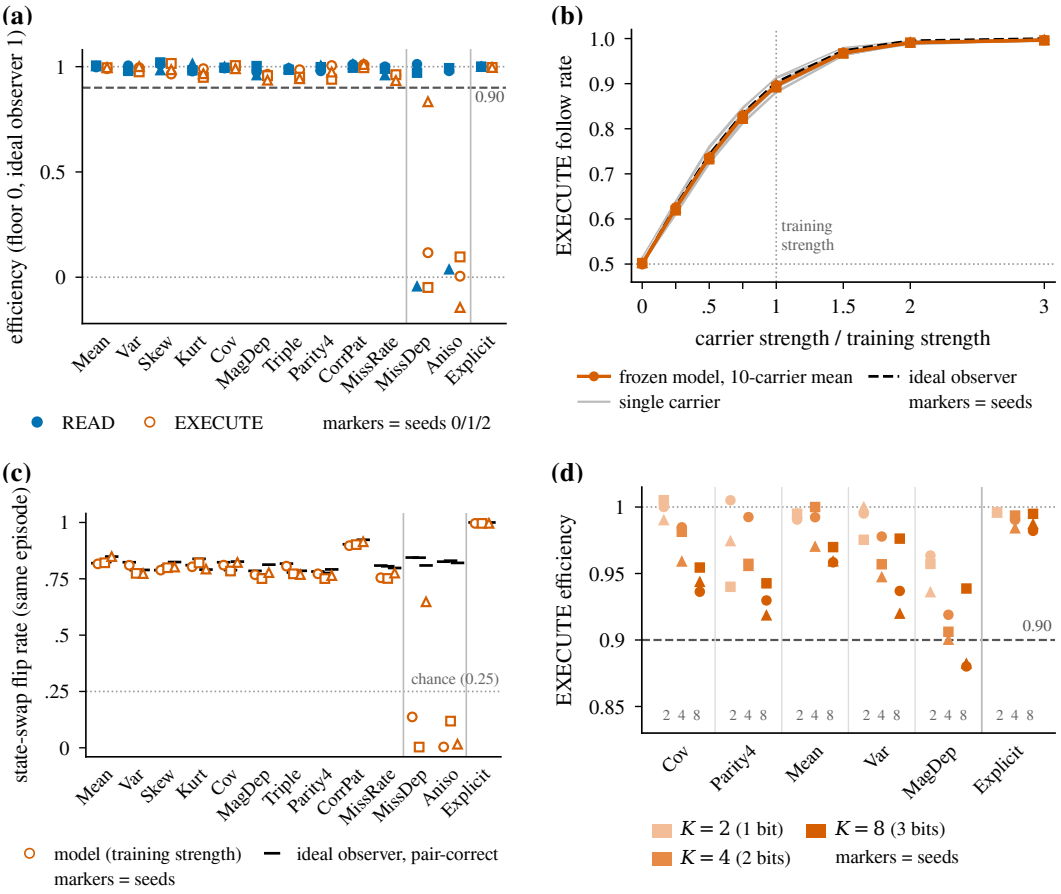



Figure 2: **Statistical structure provides writable control of frozen computation.** **(a)** Diverse carriers support READ and EXECUTE. **(b)** Control varies smoothly with carrier strength. **(c)** Counterfactual carrier rewrites change the computation of the same frozen model. **(d)** Individual carriers select among up to eight computations. Extended results are in Appendix C.

of the in-sample variation in log mean acquisition cost (Appendix C.5). *Matched decodability therefore does not imply matched acquisition.*

Interfaces can coexist, but simultaneous acquisition is not automatic. Across 32 two-carrier conditions, 20 retain both interfaces in every seed; failures concentrate in particular carriers. Once acquired, interference is small: among 163 live channel–seed measurements, intervening on the other carrier changes accuracy by at most 0.008 (Appendix C.6).

READ-type interface acquisition is also not confined to the default compact PFN: it persists from $1.08 \times 10^5$ to $4.84 \times 10^7$ parameters ($448\times$ larger) under scale-specific stability-selected optimization (Appendix C.7).

Thus, statistical states can be written, read, and executed, while multiple interfaces can coexist with little interference once acquired. How readily interfaces are acquired, however, depends strongly on their statistical structure. We next ask what assigns these interfaces their meaning.

## 3 Training Assigns Meaning to Statistical Interfaces

Section 2 establishes control, but not what gives a carrier state its meaning. One simple account is a fixed code: decode the state and dispatch its associated computation. The controlled PFN setting instead supports a probabilistic account in which context is interpreted as evidence under the training

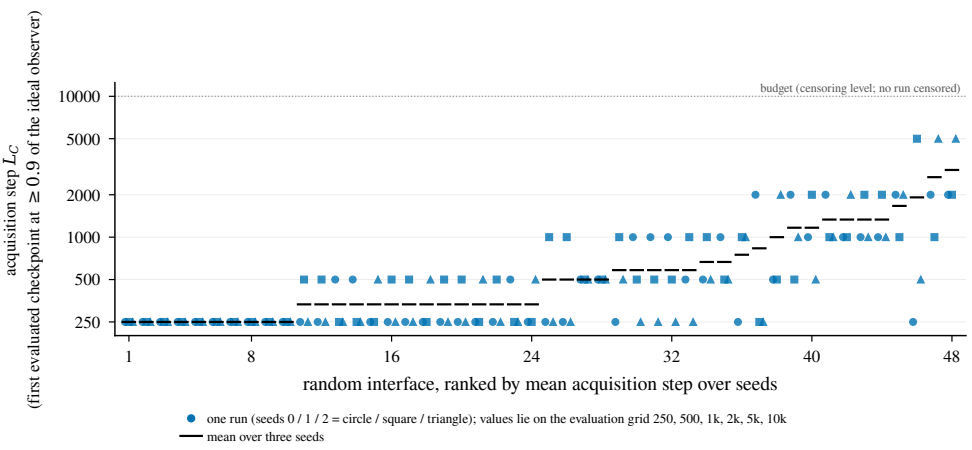



Figure 3: **Matched decodability does not imply matched acquisition.** First-crossing costs for all 48 prospectively sampled, decodability-matched interfaces across three independent seeds. All interfaces are eventually acquired, but learning time varies substantially with statistical structure.

distribution,

$$p_\theta(z \mid C) \approx p_{\text{train}}(z \mid C) \propto p_{\text{train}}(z)\, p_{\text{train}}(C \mid z). \tag{3}$$

PFNs are trained to approximate posterior prediction under their training distribution; our claim is not that this Bayesian target is itself surprising. We use the known target as a reference to ask whether a statistical property becomes a controllable interface and how its learned meaning behaves under counterfactual remapping, off-training evidence strengths, and zero-support compositions.

This yields three testable consequences: training should assign the state mapping, prior support should constrain available states, and learned likelihoods should govern the meaning of statistical evidence.

### 3.1 Prior and Likelihood Determine Interface Semantics

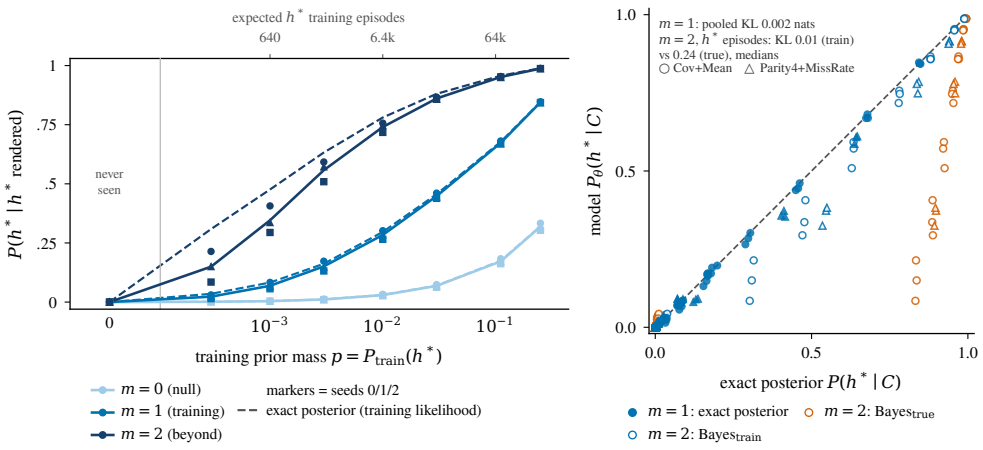
First, the mapping itself is learned. Reversing the state-to-computation mapping during training makes the same physical statistical states select opposite computations on bit-identical test inputs for covariance, magnitude dependence, and fourth-order parity (Appendix D.1).

We next vary the training prior of one joint state and, after freezing the model, independently vary carrier evidence strength. Across seven nonzero priors from $3 \times 10^{-4}$ to $0.25$, the model closely tracks the posterior implied by its training distribution; at native carrier strength, covariance plus mean yields pooled four-state $\mathrm{KL} = 0.0020$–$0.0027$ nats. The $p = 0$ condition is reported separately because both Bayes references assign zero support, making their comparison degenerate (Appendix D.2).

The stronger test moves evidence outside its training range. $\text{Bayes}_{\text{true}}$ evaluates each rendered context using its actual test-time carrier strength, whereas $\text{Bayes}_{\text{train}}$ uses the likelihood associated with the strength seen during training. For covariance plus mean, $\text{Bayes}_{\text{train}}$ is closer to the frozen model in all $105/105$ off-training cells: median absolute deviation falls from $0.0698$ to $0.0069$, and median KL from $0.2161$ to $0.0059$ nats. A second carrier family shows the same qualitative pattern (Figure 4; Appendix D.2). Carrier states therefore behave as learned evidence rather than symbols with fixed intrinsic meaning.

### 3.2 Prior Support Constrains Composition

The posterior account also predicts that individually familiar carrier states need not support a joint state absent from training. At ordinary evidence strength, withheld combinations lie far below the factorized ideal observer and instead reflect the learned joint support (Appendix D.3). With nearly unambiguous carriers, residual out-of-support recombination appears early but decreases with optimization: in a separately specified 100k trajectory for the primary high-evidence $K = 4$




(a) (b)

Figure 4: **Statistical evidence is interpreted under the training distribution.** **(a)** Model belief varies jointly with training prior and carrier evidence. **(b)** Away from the native carrier strength, frozen PFNs are better explained by the likelihood learned during training than by the likelihood implied by the physical test-time rendering.

condition, held-out joint accuracy falls from 0.468–0.544 at 10k to 0.139–0.216 at 100k, while supported-state accuracy rises from 0.983–0.988 to 0.991–0.995 (Figure 5a). The residual remains above the zero-valued support-projection reference, so this establishes a directional optimization effect rather than convergence to zero.

The boundary is graded rather than absolute: longer test-time contexts drawn from the familiar carrier distribution and different representations can recover zero-support combinations (Appendix D.3). Zero training support therefore constrains composition without constituting a general representational prohibition.

### 3.3 Multiple Interfaces Combine as Learned Evidence

Finally, we provide two conditionally independent carriers about the same latent state and vary their reliability, agreement, context size, and prior. Across 25 mean–covariance models, recomputing the fused posterior using each carrier's *training-time* likelihood reduces pooled KL from 0.06–0.32 to 0.005–0.016 nats in every model, a 5.5–45.7× reduction with median improvement of 18.9×. All 25 models fall below the 0.02-nat alignment threshold under this reference. Multiple carriers therefore combine as learned evidence rather than as symbolic commands that vote (Figure 5b; Appendix D.4).

Together, training assigns interface mappings, the training prior determines which states and combinations are supported, and learned likelihoods determine how statistical evidence is interpreted and combined. Interface semantics are therefore neither intrinsic nor freely compositional, and this training-distribution dependence persists across the 448× PFN scaling ladder (Appendix C.7).

## 4 What Can Statistical Interfaces Communicate?

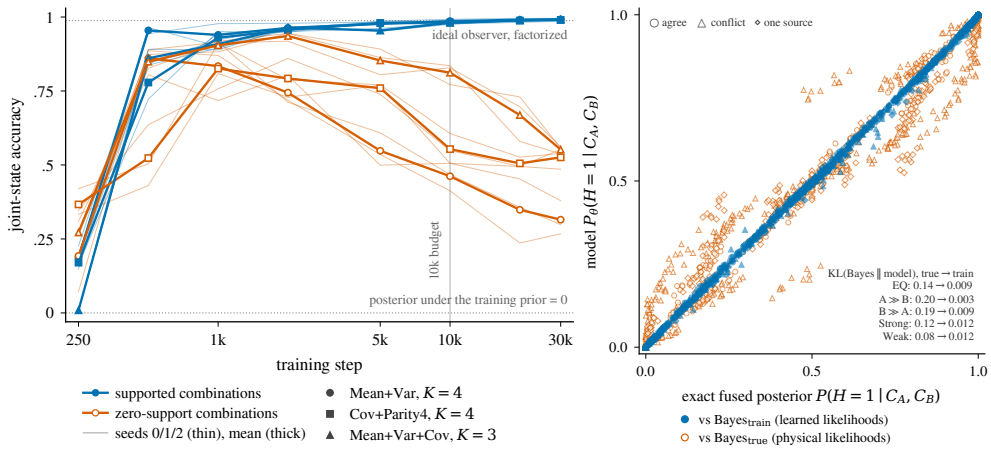
So far, statistical states have selected relatively simple behaviors. A further question is whether statistical states can communicate higher-level information that changes how a frozen model interprets ordinary observations, and whether multiple statistical states can jointly specify a structured computation.

### 4.1 Statistical Context Can Communicate a Task Prior

We construct a hierarchical task in which a latent knowledge state $K$ selects a prior $\pi_K$ over hypotheses $h$, while ordinary observations $D$ provide additional evidence:

$$p(h \mid D, K) \propto p(D \mid h)\pi_K(h).$$

The value of $K$ is communicated only through a statistical carrier and is never given explicitly.




(a) (b)

Figure 5: **Training support constrains composition, while learned likelihoods govern evidence fusion. (a)** Residual recovery of zero-support combinations decreases with continued optimization while supported performance remains high. **(b)** Two-source predictions are substantially better described by training-time carrier likelihoods than by the physical test-time rendering.

Holding $D$, the query, and the frozen model fixed, we intervene only on the carrier encoding $K$. With a near-unambiguous carrier, predictions closely match the exact knowledge-conditioned target ($\mathrm{KL} = 0.0011$–$0.0015$ nats across seeds), and rewriting $K$ produces $0.95$–$1.00$ of the corresponding exact predictive shift. With a weaker carrier, the response is attenuated in the direction expected from uncertainty about $K$ rather than behaving as if the most likely state were known with certainty (Figure 6a; Appendix E.1). Statistical context can therefore communicate a training-supported higher-level assumption that changes how otherwise identical observations are interpreted.

### 4.2 Statistical Context Can Select Structured Computations

We next compose four primitive transformations into ordered two-step programs. Two statistical carrier blocks specify the first and second instruction, yielding $4 \times 4 = 16$ possible programs while the query contains only the operand. For training-supported programs, statistical context selects and executes the requested computation at $0.89$–$0.91$ accuracy at standard carrier strength and $0.97$–$0.99$ with near-unambiguous carriers. When all 16 programs receive training support, all are executed near the decoded-program oracle.

This does not imply systematic execution of unseen programs. When four ordered combinations are withheld while every primitive remains supported in both instruction slots, statistical, explicit, and joint-codebook interfaces all remain far below the factorized reference; none of the 36 valid held-out runs satisfies the pre-specified recombination criterion (Figure 6b; Appendix E.2). Thus, statistical interfaces can communicate structured, ordered control, but they do not by themselves create systematic composition beyond the support learned during training.

## 5 Statistical Interfaces in Pretrained Language Models

The PFN experiments deliberately create statistical interfaces through the training distribution. We finally ask which related capabilities already exist in naturally pretrained models. We distinguish *sensitivity*, where changing a statistical property of demonstration context changes behavior, from *bindability*, where that property can additionally acquire a newly demonstrated behavioral meaning. These experiments test functional interfaces; they do not assume the learned-posterior mechanism established in the controlled PFNs.

### 5.1 Pretrained LLMs Respond to Statistical Structure

Sensitivity to demonstration labels, frequencies, and ordering is well documented in in-context learning (Zhao et al., 2021; Min et al., 2022; Lu et al., 2022). We therefore ask not whether such

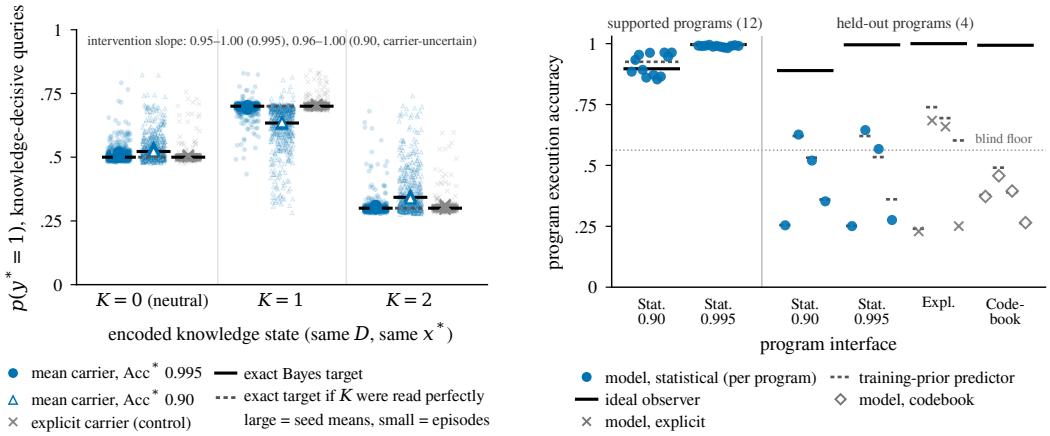



Figure 6: **Statistical interfaces can communicate knowledge and structured computation. (a)** Rewriting only the statistically encoded knowledge state changes the interpretation of otherwise identical ordinary observations according to the corresponding training-supported prior. **(b)** Statistical context reliably selects supported ordered programs, but held-out ordered compositions do not show systematic recombinational execution.

effects exist, but whether controlled statistical properties behave as manipulable interfaces. Across six frozen language models from three families, all eight tested statistical and structural properties satisfy the predefined carrier-level read criterion in every model ($48/48$ model–property combinations), although one property's effect runs systematically opposite to its predefined expected direction and is retained as a reversal. This success is not universal across candidate context structures: additional constructions testing alternation, periodicity, and grouping show little or no systematic effect. The successful properties span marginal frequencies, count and dispersion, trends across demonstrations, and relational structure, with matched controls for several simpler alternatives (Appendix F.1).

Models remain frozen and are evaluated by free generation; effects are aggregated over context seeds and evaluated against matched controls using predefined criteria (Appendix F.1).

Sensitivity is graded and multi-state. Demonstrated class frequency produces ordered responses over nine levels, and all five models tested on list count resolve eight demonstrated states from 2 to 9 at approximately one generated item per step. Two properties can also be written simultaneously: each primarily affects its associated output dimension, with median cross-talk of $0.03$ across the evaluated conditions. Natural pretraining therefore already yields graded, multi-state, and simultaneous behavioral control through statistical context (Appendix F.1).

### 5.2 Sensitivity Does Not Imply Bindability

We then ask whether a readable property can acquire an arbitrary new meaning in context. Demonstrations associate two carrier states with two output behaviors while the mapping reverses across contexts, so successful binding must follow the demonstrated mapping rather than a fixed state or answer preference. A lexical nonsense cue provides a positive control.

We expand the focused read–bind test to 12 frozen checkpoints from four model families spanning 2B to 123B parameters, including Llama 3.3 70B, Qwen3.5 122B, and Mistral Large 123B. Relational structure provides a direct dissociation: it affects behavior in all $12/12$ directly tested models yet binds in $0/12$ under the ordinary protocol, with binding effects between $0.000$ and $0.125$. Under the same protocol, block size binds in $1/12$, repetition versus diversity in $3/12$, order in $0/12$, and the lexical control in $12/12$ (Figure 7; Appendix F.2). *Statistical sensitivity is therefore not sufficient for semantic bindability under a fixed realization.*

Failure under one realization does not establish intrinsic unbindability. Among the eight models evaluated under both block-size contrasts, strengthening the carrier changes binding from $1/8$ to $8/8$ (Figure 7). We then decompose the successful carrier across nine models spanning 9B to 123B

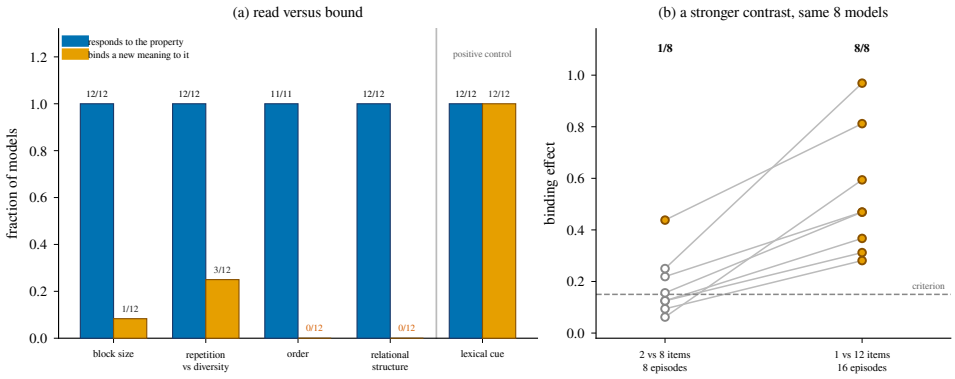



Figure 7: **Statistical sensitivity does not imply arbitrary bindability. Left:** Candidate properties are broadly readable but selectively bindable under the ordinary protocol. **Right:** Among the eight models evaluated under both block-size conditions, strengthening the physical realization changes binding from $1/8$ to $8/8$.

parameters. Approximately length-matched count manipulations bind in $9/9$ and $7/9$ models, while a count-matched large-length manipulation binds in $9/9$, ruling out simple necessity claims for either item count or block length. A tightly nuisance-matched repetition-versus-diversity construction also binds in both technically evaluable models (Appendix F.3).

These follow-ups do not identify a unique causal feature. Rather, bindability depends strongly on the physical carrier–model interaction. The reported denominators reflect the matched or technically feasible model subset for each comparison, and we did not perform analogous salience sweeps for order or relational structure. Their $0/12$ results therefore characterize the ordinary protocol rather than intrinsic unbindability.

## 6 Discussion and Limitations

Our results suggest a view of statistical structure as an *interface* rather than merely a property of data. In controlled PFNs, training can assign computational meaning to statistical properties of ordinary context, after which they become writable control variables for a frozen model. Which interfaces are acquired is not determined by available information alone, and their semantics are not intrinsic to the carrier: priors determine which states and combinations are supported, while learned likelihoods determine how evidence is interpreted and combined. Statistical interfaces can condition inference on a latent task prior or select structured computations supported by training.

Several limitations bound these conclusions. Our quantitative account of semantics is behavioral and established in controlled PFN families with known training distributions; it does not show that models internally represent priors, likelihoods, or posteriors. READ-type acquisition and evidence semantics persist across a $448\times$ PFN scaling range, but the complete EXECUTE, capacity, and coexistence stack was not repeated at scale, and larger models require scale-specific optimization. The carrier families are constructed rather than exhaustive, and acquisition, capacity, coexistence, and support boundaries can depend on representation and test regime; in particular, zero training support constrains but does not absolutely prohibit out-of-support behavior. For pretrained language models, we establish statistical sensitivity and realization-dependent bindability, but not PFN-style EXECUTE or the learned-posterior mechanism.

Statistical interfaces therefore complement rather than replace training: training establishes which computations and interface semantics are available, while inference-time structure can select or condition them without changing model parameters. This raises a broader question for frozen models: *which manipulable properties of their existing inputs have become interfaces during training?* Characterizing such latent control surfaces may provide a principled way to understand and design how pretrained models expose their computations.

## AI USE STATEMENT

Generative AI tools were used throughout the project to assist with coding, debugging, analysis, visualization, experimental development, and drafting and editing parts of the manuscript. The research questions, experimental objectives and controls, interpretation of results, and final scientific claims were determined and reviewed by the authors. The authors remained in the loop throughout the project and take full responsibility for the experiments, evidence, and manuscript. An anonymized repository accompanying the submission provides the code and experimental artifacts underlying the reported results.

## REPRODUCIBILITY STATEMENT

An anonymized repository accompanying the submission provides the code, experimental configurations, random seeds, frozen result records, and analysis utilities underlying the reported results. The reported aggregate statistics and reader-facing figures and tables can be reconstructed from these records without retraining models or repeating language-model inference, and the repository includes the information required to rerun experiments that require training or inference.

## REFERENCES

Emmanuel Abbe, Enric Boix Adsera, and Theodor Misiakiewicz. The merged-staircase property: a necessary and nearly sufficient condition for sgd learning of sparse functions on two-layer neural networks. In *Conference on Learning Theory*, pp. 4782–4887. PMLR, 2022.

Emmanuel Abbe, Enric Boix Adsera, and Theodor Misiakiewicz. Sgd learning on neural networks: leap complexity and saddle-to-saddle dynamics. In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 2552–2623. PMLR, 2023.

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alex Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RQm2KQTM5r.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=0g0X4H8yN4I.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, 36:57125–57211, 2023.

Boaz Barak, Benjamin Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit. *Advances in Neural Information Processing Systems*, 35:21750–21764, 2022.

Jan Büssing, Moritz Schlager, Timo Heiß, Tom Zehle, and Matthias Feurer. Mo-capo: Multi-objective cost-aware prompt optimization. *arXiv preprint arXiv:2605.18869*, 2026.

Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. *Advances in neural information processing systems*, 35:18878–18891, 2022.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuan-Jing Huang. Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 5997–6007, 2019.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*, 2019.

Harrison Edwards and Amos Storkey. Towards a neural statistician. *arXiv preprint arXiv:1606.02185*, 2016.

Benjamin Feuer, Robin T Schirrmeister, Valeriia Cherepanova, Chinmay Hegde, Frank Hutter, Micah Goldblum, Niv Cohen, and Colin White. Tunetables: Context optimization for scalable prior-data fitted networks. *Advances in Neural Information Processing Systems*, 37:83430–83464, 2024.

Jörg K.H. Franke, Iris K. Tennie, Michael Uhl, Ryan Koksal, Dominika Matus, Frank Hutter, Rolf Backofen, and Frederic Runge. Synthetic RNA evolution enables accurate alignment-free 3d structure prediction. In *ICLR 2026 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2026. URL https://openreview.net/forum?id=wWKuI35OW7.

Cole Gibson, Wenping Cui, and Gautam Reddy. Distinct mechanisms underlying in-context learning in transformers. *arXiv preprint arXiv:2604.12151*, 2026.

Jannik Adrian Gut, Noah Kleinschmidt, and Thomas Lemmin. Synthetic sequence alignments as programmable probes of learned conformational landscapes in deep learning protein structure predictors. *Bioinformatics*, 42(8):btag605, 2026.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333, 2023.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 3045–3059, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)*, pp. 4582–4597, 2021.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8086–8098, 2022.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.759. URL https://aclanthology.org/2022.emnlp-main.759/.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=KSugKcbNf9.

Thomas Nagler. Statistical foundations of prior-data fitted networks. In *International Conference on Machine Learning*, pp. 25660–25676. PMLR, 2023.

Allan Raventós, Mansheej Paul, Feng Chen, and Surya Ganguli. Pretraining task diversity and the emergence of non-bayesian in-context learning for regression. *Advances in neural information processing systems*, 36:14228–14246, 2023.

Arik Reuter, Tim G. J. Rudner, Vincent Fortuin, and David Rügamer. Can transformers learn full Bayesian inference in context? In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 51531–51582. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/reuter25a.html.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, 2024.

Taylor Shin, Yasaman Razeghi, Robert L Logan Iv, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 4222–4235, 2020.

Arabella Sinclair, Jaap Jumelet, Willem Zuidema, and Raquel Fernández. Structural persistence in language models: Priming as a window into abstract language representations. *Transactions of the Association for Computational Linguistics*, 10:1031–1050, 2022.

Aaditya Singh, Stephanie Chan, Ted Moskovitz, Erin Grant, Andrew Saxe, and Felix Hill. The transient nature of emergent in-context learning in transformers. *Advances in neural information processing systems*, 36:27801–27819, 2023.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron Wallace, and David Bau. Function vectors in large language models. In *International conference on learning representations*, volume 2024, pp. 17282–17333, 2024.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174. PMLR, 2023.

Hannah K Wayment-Steele, Adedolapo Ojoawo, Renee Otten, Julia M Apitz, Warintra Pitsawong, Marc Hömberger, Sergey Ovchinnikov, Lucy Colwell, and Dorothee Kern. Predicting multiple conformations via sequence clustering and alphafold2. *Nature*, 625(7996):832–839, 2024.

Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. Frequency effects on syntactic rule learning in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 932–948, 2021.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=RdJVFCHjUMI.

Haoren Xu and Guanhua Fang. Self-attention as a covariance readout: A unified view of in-context learning and repetition. *arXiv preprint arXiv:2605.10466*, 2026.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Tom Zehle. CANTANTE: Optimizing agentic systems via contrastive credit attribution. *arXiv preprint arXiv:2605.13295*, 2026.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pp. 12697–12706. Pmlr, 2021.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=92gvk82DE-.

# A Extended Related Work

Our work intersects several literatures that study how context changes the computation of a frozen model. The individual ingredients are well established: Transformers can extract statistical structure from context, infer tasks and algorithms from demonstrations, condition generation on explicit control variables, and respond to optimized prompts or interventions on internal representations. Our focus is different. We study the *interface itself*: when a manipulable statistical property of an existing input channel can be written by an external user, read by a frozen model, and executed as a change in computation. This perspective makes the origin of the interface semantics, its acquisition cost, capacity, coexistence, interference, and dependence on training support experimental objects.

**Explicit conditioning and controllable generation.** Controlling neural computation through an explicit conditioning variable has a long history in conditional generation. In text, style-transfer models condition generation on a desired attribute or style. Style Transformer (Dai et al., 2019), for example, trains a Transformer for unpaired text style transfer, while CTRL (Keskar et al., 2019) trains a language model to condition on explicit control codes governing properties such as domain, style, content, and task. Related controllable-generation methods instead guide a pretrained language model using an auxiliary attribute model, as in PPLM (Dathathri et al., 2019). These approaches establish that explicit or auxiliary conditioning signals can provide effective control over generation.

Our question begins where no dedicated control variable can be added. A statistical control interface uses a manipulable property of the model's *ordinary input* as the conditioning variable. The distinction is not one of information-theoretic expressivity: an explicit control variable could communicate the same information. Rather, it concerns where the control variable resides and how it acquires meaning. Explicit conditioning designs a control port during training; we ask when predictive training causes a property of an existing input channel to become such a port, and whether that property can subsequently be written to while the model remains frozen.

**In-context learning and contextual inference.** A large literature studies what computation Transformers perform when conditioning on demonstrations. In-context learning has been described as implicit inference over latent concepts (Xie et al., 2022), learning members of function classes from examples, and implementing learning algorithms within the forward pass (Akyürek et al., 2023; Von Oswald et al., 2023). Bai et al. (2023) further study context-dependent algorithm selection. These results establish that context can determine which predictor or computation a frozen Transformer implements.

We study a complementary question: what property of the context can carry the information that selects that computation? Rather than changing the input-output relation demonstrated by the examples, our controlled experiments intervene on statistical properties of the context and ask when these properties become writable control variables. This separates the existence of context-dependent computation from the interface through which that computation is selected.

**Training distributions, prior support, and interface semantics.** The behavior learned in context depends strongly on the pretraining distribution. Distributional properties can determine whether in-context learning emerges (Chan et al., 2022), and in-context strategies can change over the course of training (Singh et al., 2023). Task diversity can also determine whether a Transformer remains tied to a finite pretraining task distribution or generalizes beyond its support (Raventós et al., 2023). More generally, learning dynamics can depend strongly on the statistical and hierarchical structure of a target (Barak et al., 2022; Abbe et al., 2022; 2023).

These observations are central to our interpretation, but we use the training distribution in a different role. We deliberately assign meaning to a statistical carrier through its relationship with a latent state during training. This allows us to vary prior mass and evidence strength separately and compare the behavior of the frozen model with the posterior implied by its training distribution. Similarly, by matching heterogeneous carriers for ideal-observer decodability, we can distinguish information availability from the difficulty of acquiring that information as a usable interface. We do not interpret the resulting prior-support or acquisition effects as universal limits of Transformers; they characterize the controlled training distributions studied here.

**Reading statistical structure.** Transformers and related set architectures can represent population-level properties of collections of observations (Zaheer et al., 2017; Lee et al., 2019; Edwards & Storkey, 2016). More directly, recent theoretical work characterizes how attention can recover statistical structure from context. Xu and Fang relate self-attention to covariance readout (Xu & Fang, 2026), while Gibson et al. identify distinct mechanisms associated with different orders of contextual statistics (Gibson et al., 2026).

These works establish an important prerequisite for our setting: statistical structure can be accessible to the model. Accessibility alone, however, does not make a statistic a control interface. We distinguish whether a carrier is decodable, whether the model functionally reads it, whether changing it controls another computation, and whether an arbitrary behavioral meaning can be bound to it. This distinction between statistical sensitivity and statistical bindability is particularly important in our pretrained-language-model experiments.

**Prompt sensitivity and linguistic structure.** Frozen language models are sensitive to many properties of their context. Prompt format, example choice, label space, and demonstration ordering can substantially affect few-shot behavior (Zhao et al., 2021; Min et al., 2022; Lu et al., 2022). Linguistic studies likewise show effects of recent syntactic structure and distributional frequency on model predictions, including structural priming (Sinclair et al., 2022) and frequency-dependent syntactic rule learning (Wei et al., 2021).

Such observations demonstrate that context contains many variables that a pretrained model can read. We treat this as evidence for candidate interfaces, not sufficient evidence for an interface itself. Our LLM experiments therefore progress from direct sensitivity to controlled dose response and simultaneous signals, and finally to arbitrary reassignment of the behavioral meaning of a structural state. The resulting negative and positive binding results show why prompt sensitivity should not in general be equated with a writable control channel.

**Prompt, prefix, and context optimization.** A different route to controlling frozen models is to optimize their inputs. AutoPrompt searches for discrete trigger tokens (Shin et al., 2020), while prefix tuning and prompt tuning learn continuous task-specific inputs to a frozen model (Li & Liang, 2021; Lester et al., 2021). Automatic prompt-engineering methods similarly search over natural-language instructions (Zhou et al., 2023). On the PFN side, TuneTables uses prompt tuning to optimize a learned context for a pretrained PFN (Feuer et al., 2024).

These methods demonstrate the power of input-side control but differ in where the control semantics originate. Prompt and context optimization searches for a concrete input that elicits a desired response from an already trained model. Our controlled PFN experiments instead construct training distributions in which an equivalence class of contexts sharing a statistical property acquires a common meaning. Many concrete contexts can therefore instantiate the same interface state. Optimization over such properties could in turn provide a method for discovering or writing statistical interfaces automatically.

**Internal control and task representations.** A complementary literature locates control variables inside the network. In-context demonstrations can induce transferable task representations (Hendel et al., 2023), and function vectors provide causal internal representations of functions induced by demonstrations (Todd et al., 2024). Activation-steering methods directly intervene on such internal states; contrastive activation addition is one example (Rimsky et al., 2024).

This work is conceptually close to our view of frozen models as exposing controllable dimensions, but the intervention boundary differs. Activation steering identifies and modifies an internal representation. Statistical control modifies only an ordinary model input and relies on the frozen model to transform that external property into whatever internal representation implements the resulting computation. Our interfaces are therefore input-side control surfaces and require neither parameter changes nor direct access to hidden activations.

**PFNs as a controlled laboratory.** Prior-Data Fitted Networks are trained on samples from a synthetic prior to amortize prediction under that distribution (Müller et al., 2022). Subsequent work develops their statistical foundations (Nagler, 2023) and extends prior-fitted inference to richer Bayesian posterior-inference problems (Reuter et al., 2025). This makes PFNs particularly useful for

our purposes because both the data-generating distribution and the corresponding inference target can be specified explicitly.

We use this property not to argue that Transformers generally implement Bayesian inference, but to create a setting in which the semantics of a contextual interface are known. A carrier provides evidence about a latent state under the training distribution, while the training prior determines the prior mass of that state. We can therefore compare a frozen model quantitatively with the posterior implied by the distribution on which it was trained and intervene independently on prior support, carrier evidence, and the context presented at inference time.

**Biological context and multiple-sequence alignments.** Biological sequence models provide a separate setting in which structured population context can influence predictions. In structure prediction, manipulating multiple-sequence alignments can alter the output of frozen predictors. Sequence clustering has been used to elicit alternative protein conformations (Wayment-Steele et al., 2024). Structure-informed synthetic alignments can improve RNA structure prediction (Franke et al., 2026), while synthetic alignments designed to encode alternative structural constraints can steer protein structure predictors toward distinct conformational states (Gut et al., 2026).

These results provide an important input-side precedent: structured context supplied through an existing model input can causally alter a frozen predictor. We treat them as motivation and convergent evidence rather than as evidence that biological structure predictors implement the same mechanism as our controlled PFNs. The PFN experiments allow us to isolate questions that are difficult to answer in a naturally trained structure model: what gives a carrier its meaning, how evidence interacts with prior support, which statistical properties are acquired, how several interfaces coexist, and when their states can be composed.

**Relation to neighboring forms of control.** The neighboring approaches differ primarily in where control information is represented and where its semantics originate. Explicit conditional models provide a dedicated control variable during training. Prompt and prefix methods place control in optimized tokens or vectors. Activation steering intervenes on internal representations. In-context learning infers tasks or algorithms from demonstrations. Biological MSA interventions manipulate an existing structured input. Our object of study is narrower: a manipulable statistical property of ordinary context whose state is read by a frozen model and causally controls its subsequent computation.

This distinction is not intended to position statistical control as a replacement for explicit conditioning, prompting, or fine-tuning. Explicit control is preferable whenever an appropriate control port can be designed. The statistical-interface question becomes relevant when the model is frozen and its existing input channel is the available route for control. Our goal is therefore to characterize when such an implicit interface exists, how it acquires meaning, and what constrains its use.

# B Controlled PFN Testbed and Statistical Interfaces

This appendix specifies the controlled PFN testbed used in Section 2: the model and training setup, the distinction between READ and EXECUTE, the hand-designed statistical carriers, finite-context calibration, and the nuisance-information audit. Extended results for the experiments of Section 2 are reported in Appendix C.

## B.1 PFN Architecture and Training

All controlled experiments use transformer-based Prior-Data Fitted Networks. Unless stated otherwise, the default backbone has model dimension $64$, three transformer layers, four attention heads, and feed-forward dimension $128$. The transformer is permutation-equivariant over context rows: it uses neither row-position encodings nor rotary positional embeddings. The default binary models contain approximately $1.1 \times 10^5$ trainable parameters.

Contexts contain $N = 128$ rows by default. A carrier occupies a four-column block consisting of its informative variables and independent standard-normal padding where required. Query rows contain fresh draws in carrier columns and therefore do not reveal the carrier state directly. For EXECUTE, the query additionally contains an operand. Training uses AdamW with gradient clipping, linear

warm-up over the first $5\%$ of optimization, and cosine learning-rate decay. The default budget is $10{,}000$ optimization steps with $64$ episodes per step; specified extended-budget experiments use $30{,}000$ steps. Unless stated otherwise, experiments use three independently initialized model seeds (Table 1).

Table 1: **Default controlled-PFN configuration.** Experiment-specific deviations are stated with the corresponding experiment.

| Component | Default setting |
|---|---|
| Context size | $N = 128$ |
| Model dimension | $64$ |
| Transformer layers | $3$ |
| Attention heads | $4$ |
| Feed-forward dimension | $128$ |
| Default parameters | $\approx 1.1 \times 10^5$ |
| Optimizer | AdamW |
| Peak learning rate | $10^{-3}$ |
| Weight decay | $0$ |
| Gradient clipping | $1.0$ |
| Schedule | $5\%$ linear warm-up, cosine decay |
| Default training budget | $10{,}000 \times 64$ episodes |
| Default model seeds | $0, 1, 2$ |
| Evaluation checkpoints | $250, 500, 1000, 2000, 5000, 10000$ steps |

The released reproducibility package pins the exact software dependencies and contains the frozen configurations and seeds used by each run.

### B.2 READ, EXECUTE, AND CONTROLS

Let $z \in \{0, \ldots, K-1\}$ denote the latent state encoded by a carrier in context $C$.

In READ, the model predicts $z$ at a query row whose own carrier features contain no state information. A state-blind predictor therefore achieves accuracy $1/K$.

In EXECUTE, the target is not the state itself. Instead, $z$ selects one operation from a fixed Sylvester–Hadamard codebook, and the model must apply that operation to an operand $x^*$ supplied only at the query:

$$\mathrm{op}_z(x^*) = \frac{1}{2}\left(1 - H_{z+1,\lfloor \Phi(x^*)M \rfloor}\right), \tag{4}$$

where $H$ is a Hadamard matrix and $M$ is the smallest admissible power of two. Thus, EXECUTE requires the context state to affect a separate computation rather than merely be reported. For $K = 2$, the exact state-blind optimum of the downstream task is $0.75$. Table 2 summarizes the tasks and their paired controls.

Table 2: **Controlled tasks and interpretation.**

| Task | Target | State-blind baseline | Successful behavior shows |
|---|---|---|---|
| READ | $z$ | $1/K$ | carrier state is accessible |
| EXECUTE | $\mathrm{op}_z(x^*)$ | task-specific | carrier controls computation |
| DUAL-READ | addressed carrier state | $1/K$ | multiple states coexist |

Every trained model is also evaluated under matched controls. In the *broken* condition, the carrier is re-rendered from the same base randomness using an independently sampled state while the target remains tied to the original state. The *no-information* condition renders the carrier at its state-free null. An *explicit* carrier writes the state noiselessly into a dedicated input feature and provides an upper reference for accessibility. Finally, the `row_order` control encodes state only through the order of an otherwise identical multiset of rows; because the PFN is permutation equivariant over context rows, this information is unavailable to the model by construction.

### B.3 Statistical Carrier Construction

A carrier is a state-conditional distribution over context rows,

$$C \sim p_{\mathcal{T}}(C \mid z), \tag{5}$$

whose latent state changes designated statistical structure. The downstream task determines what this state means computationally. For many constructions, nuisance quantities are matched analytically; for others, narrower matching claims are checked empirically in Appendix B.5.

The initial atlas contains 14 statistical carriers plus two controls. They span marginal moments, pairwise and higher-order dependence, missingness, geometry, and dependence across context rows. Tables 3 and 4 group them by the kind of statistic that carries the state and record, for each carrier, which quantities the construction holds fixed between states. Two further carriers enter only the coexistence screen and are listed with the group they belong to.

Table 3: **Marginal and distributional carriers.** The state is carried by a property of the single-column or single-row distribution. CLEAN denotes carriers retained in audit-clean aggregate claims; LEAK denotes carriers retained for qualified per-carrier analysis but excluded from those aggregates; COEXIST. denotes a carrier used only in the coexistence screen.

| Carrier | State-bearing statistic | Held fixed between states | Status |
|---|---|---|---|
| `mean` | column mean | nothing beyond the base noise model | CLEAN |
| `variance` | column variance | the marginal is deliberately not matched, since it is the property itself | CLEAN |
| `skewness` | standardized third moment | mean 0 and variance 1 exactly; the two states are mirror images, so kurtosis matches too | CLEAN |
| `kurtosis` | excess kurtosis | mean 0, variance 1, skewness 0 exactly | CLEAN |
| `tail_shape` | far-tail mass at fixed kurtosis | mean 0, variance 1, skewness 0 and excess kurtosis exactly; sixth moments differ | COEXIST. |
| `missing_rate` | marginal missingness rate | the distribution of observed values; the zero count necessarily co-varies with column variance | CLEAN |
| `anisotropy` | eigenvalue gap of the row covariance | $E[\Sigma] = I$ under every state; no fixed column statistic is informative in expectation | CLEAN |
| `cluster_sep` | cluster separation | $E[x] = 0$ and $\mathrm{Cov}(x) = I$ exactly; total variance and effective rank | LEAK |

Table 4: **Dependency and structural carriers, and the two controls.** The state is carried by a relation between columns, between rows, or between missingness masks, so no single marginal identifies it. Status labels are as in Table 3.

| Carrier | State-bearing statistic | Held fixed between states | Status |
|---|---|---|---|
| `covariance` | sign of the correlation | both marginals exactly $\mathcal{N}(0,1)$; $\lvert\rho\rvert$ and the spectrum equal across states | CLEAN |
| `binary_cov` | correlation of two binary columns | marginals exactly uniform; column moments, histograms, $\lvert\mathrm{corr}\rvert$ and spectrum | COEXIST. |
| `magnitude_dep` | dependence between magnitudes | $E[uv] = 0$ exactly and both marginals exactly $\mathcal{N}(0,1)$ | CLEAN |
| `triple_interaction` | third-order interaction | all three pairwise covariances 0; the third column's marginal is identical across states | CLEAN |
| `parity4` | fourth-order parity | every mean, pairwise product and triple product has mean 0 exactly | CLEAN |
| `corr_pattern` | pattern of correlation signs | unit diagonal, $\lvert\mathrm{corr}\rvert$, eigenspectrum and total variance identical across states | CLEAN |
| `missing_dep` | dependence between missingness masks | each column missing with probability exactly $p$; observed values identically distributed | CLEAN |
| `row_dependence` | dependence between context rows | each row exactly $\mathcal{N}(0, I_2)$; every column statistic and the single-row empirical distribution | LEAK |
| `explicit` | the value of a dedicated column | nothing; this positive control claims no matching | CONTROL |
| `row_order` | the order of otherwise identical rows | everything: the two states share an identical multiset of rows | CONTROL |

Exact state-conditional generators and their frozen strengths are included in the released experimental registry and implementation.

### B.4 Finite-Context Ideal Observers and Calibration

Native carrier parameters are not directly comparable across statistical constructions. We therefore calibrate carriers by the information about their latent state available from a finite context. For a carrier with $K$ states, the ideal observer is

$$\hat{z}(C) = \arg \max_{z \in \{0,\ldots,K-1\}} \log p(C \mid z), \tag{6}$$

with finite-context accuracy

$$\mathrm{Acc}^*(N) = \Pr[\hat{z}(C) = z]. \tag{7}$$

For the default binary experiments, carrier strength is chosen to target

$$\mathrm{Acc}^*(128) \approx 0.90. \tag{8}$$

For independent-row constructions, the context log likelihood is the sum of the exact row log likelihoods. The `anisotropy` and `cluster_sep` constructions marginalize an episode-level latent orientation numerically. The `row_dependence` construction uses a fixed mean-field pairing observer and is therefore treated separately.

Calibration equalizes finite-context distinguishability, not the physical magnitude of the carrier parameter. Nor does matched ideal-observer accuracy imply that every named statistic $T(C)$ is formally sufficient for the latent state. Exact sufficiency holds when the state-dependent context distribution differs only through $T(C)$; elsewhere we use finite-context decodability as the operational information measure.

Carrier strengths are fixed before PFN training outcomes are observed. The default atlas calibration uses predefined strength brackets and Monte Carlo acceptance around the $0.90$ target; the random-interface experiment of Appendix C.4 uses the same principle with its own predefined tolerance and search bracket.

### B.5 Carrier Confound Audit

Matched ideal-observer accuracy does not guarantee that a carrier changes only the statistic it was designed to encode. Finite samples may expose unintended state-dependent structure that a model could exploit. We therefore audit each hand-designed carrier at its frozen calibrated strength before using it in audit-clean aggregate claims.

For each carrier, independent training and held-out contexts are summarized by families of marginal moments, histograms, missingness statistics, covariance and correlation quantities, eigenvalue summaries, higher-order cross-statistics, projected kurtosis, nearest-neighbour distances, and related diagnostics. Standardized logistic probes predict carrier state from each statistic family. For a carrier whose construction claims a set of statistics should remain matched, we define the recovered nuisance fraction

$$f_{\text{nuis}} = \frac{a_{\text{nuis}} - a_0}{a^* - a_0}, \tag{9}$$

where $a_{\text{nuis}}$ is the strongest held-out nuisance-probe accuracy, $a_0$ is chance, and $a^*$ is the independently measured ideal-observer accuracy.

Twelve of the fourteen statistical atlas carriers pass this audit. Two do not. For `cluster_sep`, nuisance probes recover approximately $62\%$ of the available above-chance signal, primarily through marginal-shape statistics. For `row_dependence`, they recover approximately $32\%$, largely through finite-sample covariance fluctuations. Both carriers remain available for qualified analyses but are excluded from the audit-clean headline population (Table 5).

Table 5: **Atlas carriers excluded from audit-clean aggregate claims.**

| Carrier | Intended structure | strongest nuisance pathway | recovered signal |
|---|---|---|---|
| `cluster_sep` | cluster geometry | marginal shape / kurtosis | $62.4\%$ |
| `row_dependence` | dependence across rows | covariance / eigenvalue summaries | $31.7\%$ |

## C Extended Results: Writable Statistical Interfaces

This appendix provides the detailed evidence supporting Sections 2.2 and 2.3: performance across the hand-designed carriers, counterfactual rewriting, graded and multi-state control, the prospectively sampled structural-interface family, coexistence, scaling, and observed boundaries.

### C.1 READ and EXECUTE Across Hand-Designed Carriers

Because carriers differ slightly in their realized finite-context distinguishability, we summarize performance relative to the corresponding oracle:

$$\eta = \frac{a - a_{\text{floor}}}{a_{\text{oracle}} - a_{\text{floor}}}, \tag{10}$$

where $a$ is model accuracy and the floor is the appropriate state-blind baseline. For READ, $a_{\text{floor}} = 1/K$. For binary EXECUTE, the correct state-blind task floor is $0.75$.

At the default $10{,}000$-step budget, 10 of the 12 audit-clean carriers reach $\eta \geq 0.90$ in every one of three seeds on both separately trained READ and EXECUTE models. These carriers span marginal moments, second- and higher-order dependencies, and missingness. The $\eta \geq 0.90$ threshold is a descriptive near-oracle summary rather than a training or selection criterion.

The carrier- and seed-level measurements below show that the headline result is not driven by a single statistical family.

Table 6: Seed-level READ and EXECUTE results of the 16 atlas carriers ($K = 2$, $N = 128$, 10 000 steps, final checkpoint, 1 000 evaluation episodes per run). Entries are efficiencies $(\mathrm{acc} - \mathrm{floor})/(\mathrm{oracle} - \mathrm{floor})$ with floor $1/K$ (READ) or the state-blind Hadamard floor 0.75 (EXECUTE) and the ideal observer on the same episodes (READ: state accuracy; EXECUTE: accuracy of acting on the decoded state); the count criterion is efficiency $\geq 0.90$ in every seed on both tasks.

| carrier | audit | $\mathrm{Acc}^*$ | $\mathrm{oracle}_X$ | $R_0$ | $R_1$ | $R_2$ | $X_0$ | $X_1$ | $X_2$ | status at 10k |
|---|---|---|---|---|---|---|---|---|---|---|
| mean | CLEAN | 0.918 | 0.958 | 1.00 | 1.00 | 1.00 | 0.99 | 0.99 | 1.00 | **pass** (both tasks, 3/3 seeds) |
| variance | CLEAN | 0.906 | 0.953 | 1.00 | 0.98 | 0.99 | 1.00 | 0.98 | 1.00 | **pass** (both tasks, 3/3 seeds) |
| skewness | CLEAN | 0.905 | 0.952 | 1.01 | 1.02 | 0.98 | 0.97 | 1.02 | 0.99 | **pass** (both tasks, 3/3 seeds) |
| kurtosis | CLEAN | 0.908 | 0.953 | 0.98 | 0.98 | 1.01 | 0.99 | 0.95 | 0.97 | **pass** (both tasks, 3/3 seeds) |
| covariance | CLEAN | 0.899 | 0.949 | 0.99 | 0.99 | 1.00 | 1.00 | 1.01 | 0.99 | **pass** (both tasks, 3/3 seeds) |
| magnitude_dep | CLEAN | 0.887 | 0.944 | 0.97 | 1.00 | 0.96 | 0.96 | 0.96 | 0.94 | **pass** (both tasks, 3/3 seeds) |
| triple_interaction | CLEAN | 0.905 | 0.954 | 0.99 | 0.99 | 0.99 | 0.99 | 0.95 | 0.94 | **pass** (both tasks, 3/3 seeds) |
| parity4 | CLEAN | 0.897 | 0.948 | 0.98 | 1.00 | 1.01 | 1.01 | 0.94 | 0.97 | **pass** (both tasks, 3/3 seeds) |
| corr_pattern | CLEAN | 0.900 | 0.948 | 1.01 | 1.00 | 1.01 | 1.01 | 1.00 | 1.01 | **pass** (both tasks, 3/3 seeds) |
| missing_rate | CLEAN | 0.904 | 0.952 | 1.00 | 0.99 | 0.96 | 0.96 | 0.96 | 0.93 | **pass** (both tasks, 3/3 seeds) |
| missing_dep | CLEAN | 0.918 | 0.957 | 1.01 | 0.97 | -0.04 | 0.12 | -0.05 | 0.83 | seed-split: READ 2/3, EXECUTE 0/3 |
| anisotropy | CLEAN | 0.891 | 0.946 | 0.98 | 0.99 | 0.04 | 0.00 | 0.10 | -0.14 | seed-split: READ 2/3, EXECUTE 0/3 |
| cluster_sep | LEAK | 0.887 | 0.943 | 0.96 | 0.99 | 0.98 | 0.81 | 0.90 | 0.90 | LEAK: excluded from the count |
| row_dependence | LEAK | 0.894 | 0.950 | 0.93 | 0.89 | 0.86 | 0.08 | 0.02 | 0.30 | LEAK: excluded from the count |
| explicit | control | 1.000 | 1.000 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | control (upper reference) |
| row_order | control | 1.000 | 1.000 | -0.05 | 0.03 | -0.03 | 0.02 | -0.05 | -0.05 | control (unavailable; at chance as designed) |
| missing_dep (30k steps) | CLEAN | 0.918 | 0.957 | 0.99 | 0.99 | 0.99 | 0.97 | -0.08 | 0.99 | extended 30,000-step budget: not part of the 10,000-step count |
| anisotropy (30k steps) | CLEAN | – | 0.946 | – | – | – | 0.02 | -0.01 | -0.06 | extended 30,000-step budget: not part of the 10,000-step count |
| row_dependence (30k steps) | LEAK | – | 0.950 | – | – | – | 0.77 | 0.01 | 0.95 | extended 30,000-step budget: not part of the 10,000-step count |

$\mathrm{Acc}^*$ = READ ideal-observer accuracy, $\mathrm{oracle}_X$ = EXECUTE task oracle (both averaged over the three runs' evaluation episodes); $R_s$ / $X_s$ = READ / EXECUTE efficiency of the seed-$s$ model. Groups: the ten audit-clean carriers that pass; the two audit-clean carriers whose result is seed-split; the two LEAK carriers (audit, Appendix B.5), never counted; the two controls. Raw accuracies, confidence intervals, broken / no-information / permuted conditions and run identifiers are in the accompanying CSV. The pre-specified pass criterion (mean accuracy $\geq$ chance $+0.10$, valid$-$broken CI $> 0$ in every seed, controls within 0.05 of chance) is satisfied by every carrier except missing_dep, anisotropy, row_dependence (EXECUTE) and row_order. Extended-budget rows are separate runs at 30 000 steps and answer a different question (whether a seed-split cell is a budget effect); they do not enter the count.

### C.2 Counterfactual Rewriting and Graded Control

Reading a state does not by itself establish a writable interface. We therefore evaluate each frozen model on matched counterfactual context pairs that share base randomness, padding, query rows, and query operand while the carrier state alone is rewritten.

A pair is counted as correct only when the model follows the written state under both renders. Among the ten audit-clean carriers in the headline intervention set, nine satisfy the complete pre-specified overwrite criterion in all three seeds. At the training strength their pair-correct rates are approximately 0.75–0.85, corresponding to 0.93–1.00 of the ideal-observer pair rate on the same episodes.

The exception is corr_pattern. Its rewrite itself is strong, but its nominal zero-strength construction does not eliminate all information because the paired sign patterns remain exact flips. It therefore fails the null component of the complete intervention criterion; this is a limitation of that null construction rather than a failure to change model behavior by rewriting the carrier (Figure 8).

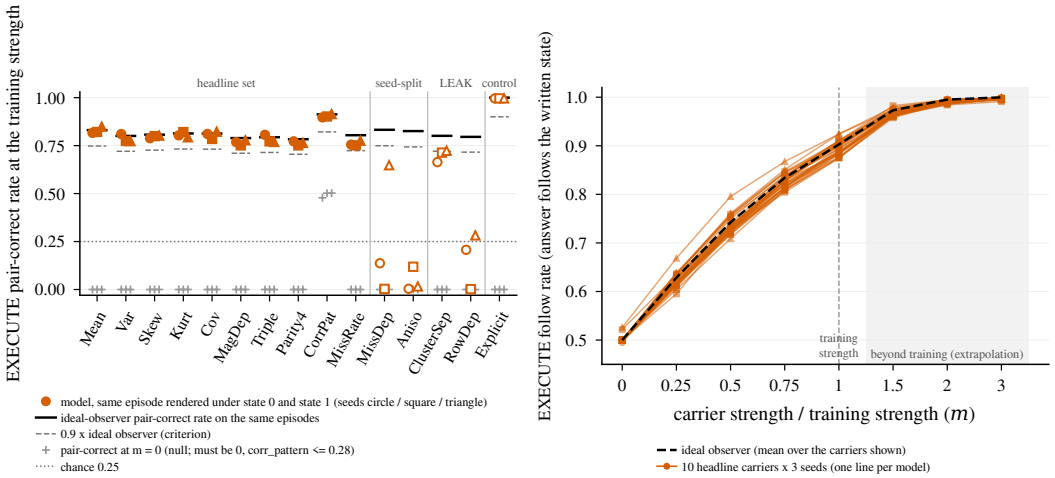



Figure 8: **Counterfactual and graded control. (a)** Matched carrier rewrites change the computation of the same frozen EXECUTE model. **(b)** EXECUTE behavior varies monotonically as carrier strength is changed at inference time; $m = 1$ is the training strength.

For graded control, carrier strength is multiplied at inference time by

$$m \in \{0, 0.25, 0.5, 0.75, 1, 1.5, 2, 3\}.$$

Models are trained only at $m = 1$. Across the ten headline carriers and three seeds, the tested READ and EXECUTE curves satisfy the graded-control criterion. Averaged across headline carriers, state-following increases from $0.501$ at $m = 0$ to $0.896$ at $m = 1$, and to $0.968$, $0.991$, and $0.997$ at $m = 1.5, 2, 3$, respectively. The corresponding ideal-observer values are $0.500$, $0.903$, $0.973$, $0.995$, and $1.000$.

Table 7: Paired counterfactual rewrite at the training strength ($m = 1$): for each frozen model, 2000 episodes (decisive queries only for EXECUTE) are rendered under state 0 and, with identical base randomness, query rows and operand, under state 1; pair-correct = the answer follows the written state under both renders (chance 0.25). Oracle pair = the same rate for the ideal observer on the same episodes. Pre-specified overwrite criterion: pair-correct $\geq 0.9\times$ oracle pair and pair-correct $= 0$ at $m = 0$ (identical tensors; $\leq 0.28$ for corr_pattern whose $m = 0$ renders are sign flips of the same noise).

| carrier | audit | seed | EXEC pair-correct | oracle pair | ratio | pair at $m$=0 | READ pair-correct | oracle pair | ratio | overwrite criterion (EXECUTE) |
|---|---|---|---|---|---|---|---|---|---|---|
| mean | CLEAN | 0 | 0.816 | 0.819 | 1.00 | 0.00 | 0.822 | 0.822 | 1.00 | headline: pass |
| | | 1 | 0.821 | 0.824 | 1.00 | 0.00 | 0.820 | 0.822 | 1.00 | headline: pass |
| | | 2 | 0.849 | 0.849 | 1.00 | 0.00 | 0.841 | 0.840 | 1.00 | headline: pass |
| variance | CLEAN | 0 | 0.810 | 0.823 | 0.98 | 0.00 | 0.808 | 0.807 | 1.00 | headline: pass |
| | | 1 | 0.774 | 0.789 | 0.98 | 0.00 | 0.781 | 0.784 | 1.00 | headline: pass |
| | | 2 | 0.773 | 0.789 | 0.98 | 0.00 | 0.790 | 0.794 | 0.99 | headline: pass |
| skewness | CLEAN | 0 | 0.788 | 0.788 | 1.00 | 0.00 | 0.797 | 0.793 | 1.00 | headline: pass |
| | | 1 | 0.799 | 0.808 | 0.99 | 0.00 | 0.804 | 0.804 | 1.00 | headline: pass |
| | | 2 | 0.802 | 0.824 | 0.97 | 0.00 | 0.812 | 0.819 | 0.99 | headline: pass |
| kurtosis | CLEAN | 0 | 0.803 | 0.811 | 0.99 | 0.00 | 0.803 | 0.808 | 0.99 | headline: pass |
| | | 1 | 0.821 | 0.838 | 0.98 | 0.00 | 0.816 | 0.824 | 0.99 | headline: pass |
| | | 2 | 0.793 | 0.791 | 1.00 | 0.00 | 0.812 | 0.805 | 1.01 | headline: pass |
| covariance | CLEAN | 0 | 0.811 | 0.823 | 0.98 | 0.00 | 0.808 | 0.808 | 1.00 | headline: pass |
| | | 1 | 0.785 | 0.789 | 0.99 | 0.00 | 0.808 | 0.807 | 1.00 | headline: pass |
| | | 2 | 0.822 | 0.826 | 1.00 | 0.00 | 0.817 | 0.818 | 1.00 | headline: pass |
| magnitude_dep | CLEAN | 0 | 0.768 | 0.785 | 0.98 | 0.00 | 0.775 | 0.789 | 0.98 | headline: pass |
| | | 1 | 0.751 | 0.769 | 0.98 | 0.00 | 0.761 | 0.773 | 0.98 | headline: pass |
| | | 2 | 0.778 | 0.813 | 0.96 | 0.00 | 0.781 | 0.794 | 0.98 | headline: pass |
| triple_interaction | CLEAN | 0 | 0.806 | 0.817 | 0.99 | 0.00 | 0.806 | 0.817 | 0.99 | headline: pass |
| | | 1 | 0.772 | 0.778 | 0.99 | 0.00 | 0.780 | 0.780 | 1.00 | headline: pass |
| | | 2 | 0.769 | 0.785 | 0.98 | 0.00 | 0.784 | 0.798 | 0.98 | headline: pass |
| parity4 | CLEAN | 0 | 0.772 | 0.782 | 0.99 | 0.00 | 0.775 | 0.777 | 1.00 | headline: pass |
| | | 1 | 0.751 | 0.776 | 0.97 | 0.00 | 0.783 | 0.797 | 0.98 | headline: pass |
| | | 2 | 0.764 | 0.792 | 0.97 | 0.00 | 0.786 | 0.797 | 0.99 | headline: pass |
| corr_pattern | CLEAN | 0 | 0.897 | 0.903 | 0.99 | 0.48 | 0.897 | 0.902 | 0.99 | headline: fails null check (m=0 pair-correct 0.48) |
| | | 1 | 0.902 | 0.912 | 0.99 | 0.50 | 0.900 | 0.904 | 1.00 | headline: fails null check (m=0 pair-correct 0.50) |
| | | 2 | 0.914 | 0.923 | 0.99 | 0.50 | 0.911 | 0.914 | 1.00 | headline: fails null check (m=0 pair-correct 0.50) |
| missing_rate | CLEAN | 0 | 0.755 | 0.809 | 0.93 | 0.00 | 0.793 | 0.807 | 0.98 | headline: pass |
| | | 1 | 0.752 | 0.806 | 0.93 | 0.00 | 0.798 | 0.817 | 0.98 | headline: pass |
| | | 2 | 0.775 | 0.798 | 0.97 | 0.00 | 0.779 | 0.800 | 0.97 | headline: pass |
| missing_dep | CLEAN | 0 | 0.137 | 0.844 | 0.16 | 0.00 | 0.825 | 0.830 | 0.99 | diagnostic: no |
| | | 1 | 0.003 | 0.844 | 0.00 | 0.00 | 0.827 | 0.845 | 0.98 | diagnostic: no |
| | | 2 | 0.648 | 0.809 | 0.80 | 0.00 | 0.001 | 0.816 | 0.00 | diagnostic: no |
| anisotropy | CLEAN | 0 | 0.004 | 0.826 | 0.00 | 0.00 | 0.829 | 0.832 | 1.00 | diagnostic: no |
| | | 1 | 0.118 | 0.830 | 0.14 | 0.00 | 0.831 | 0.826 | 1.01 | diagnostic: no |
| | | 2 | 0.016 | 0.821 | 0.02 | 0.00 | 0.000 | 0.820 | 0.00 | diagnostic: no |
| cluster_sep | LEAK | 0 | 0.664 | 0.796 | 0.83 | 0.00 | 0.771 | 0.797 | 0.97 | diagnostic: no |
| | | 1 | 0.713 | 0.802 | 0.89 | 0.00 | 0.781 | 0.798 | 0.98 | diagnostic: no |
| | | 2 | 0.723 | 0.805 | 0.90 | 0.00 | 0.776 | 0.799 | 0.97 | diagnostic: no |
| row_dependence | LEAK | 0 | 0.207 | 0.794 | 0.26 | 0.00 | 0.720 | 0.766 | 0.94 | diagnostic: no |
| | | 1 | 0.002 | 0.799 | 0.00 | 0.00 | 0.688 | 0.795 | 0.87 | diagnostic: no |
| | | 2 | 0.282 | 0.794 | 0.36 | 0.00 | 0.698 | 0.785 | 0.89 | diagnostic: no |
| explicit | NO CLAIMS (control) | 0 | 0.996 | 1.000 | 1.00 | 0.00 | 1.000 | 1.000 | 1.00 | diagnostic: pass |
| | | 1 | 0.996 | 1.000 | 1.00 | 0.00 | 1.000 | 1.000 | 1.00 | diagnostic: pass |
| | | 2 | 0.996 | 1.000 | 1.00 | 0.00 | 1.000 | 1.000 | 1.00 | diagnostic: pass |

Headline set = the ten audit-clean carriers that passed the pre-specified atlas criterion, fixed before this evaluation; the remaining carriers are evaluated identically and reported as diagnostics. row_order has no strength parameter and no dose axis. Run identifiers and confidence intervals are in the CSV.

Table 8: Graded strength response of the frozen models: follow rate (answer follows the written state, mean over the two states, decisive queries) at multipliers $m$ of the calibrated training strength for EXECUTE ($X_m$) and READ ($R_m$) checkpoints, per seed; 2000 episodes with identical base randomness across all $m$. A = pre-specified criterion flags: M/m monotone over $m \leq 1$ (no drop $> 0.04$), G/g graded ($f(0.5) \geq f(0)+0.05$ and $\leq f(1)-0.05$), C/c no collapse ($f(2) \geq f(1)-0.03$; – if $m = 2$ is outside the construction's domain); upper case = holds.

| carrier | seed | $X_0$ | $X_{0.5}$ | $X_1$ | $X_2$ | $X_3$ | A | $R_0$ | $R_{0.5}$ | $R_1$ | $R_2$ | $R_3$ | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| mean | 0 | 0.50 | 0.76 | 0.91 | 0.99 | 0.99 | MGC | 0.50 | 0.75 | 0.91 | 1.00 | 1.00 | MGC |
| | 1 | 0.50 | 0.75 | 0.91 | 1.00 | 1.00 | MGC | 0.50 | 0.75 | 0.91 | 1.00 | 1.00 | MGC |
| | 2 | 0.50 | 0.76 | 0.92 | 0.99 | 1.00 | MGC | 0.50 | 0.75 | 0.92 | 1.00 | 1.00 | MGC |
| variance | 0 | 0.50 | 0.75 | 0.90 | 1.00 | 1.00 | MGC | 0.50 | 0.74 | 0.90 | 1.00 | 1.00 | MGC |
| | 1 | 0.50 | 0.74 | 0.89 | 0.98 | 0.99 | MGC | 0.50 | 0.74 | 0.89 | 0.99 | 1.00 | MGC |
| | 2 | 0.50 | 0.74 | 0.89 | 0.99 | 0.99 | MGC | 0.50 | 0.74 | 0.89 | 0.99 | 1.00 | MGC |
| skewness | 0 | 0.50 | 0.74 | 0.89 | 0.99 | 1.00 | MGC | 0.50 | 0.74 | 0.90 | 0.99 | 1.00 | MGC |
| | 1 | 0.50 | 0.73 | 0.90 | 0.99 | 1.00 | MGC | 0.50 | 0.74 | 0.90 | 0.99 | 1.00 | MGC |
| | 2 | 0.50 | 0.76 | 0.90 | 0.99 | 0.99 | MGC | 0.50 | 0.76 | 0.91 | 0.99 | 1.00 | MGC |
| kurtosis | 0 | 0.50 | 0.72 | 0.90 | 0.99 | 1.00 | MGC | 0.50 | 0.72 | 0.90 | 0.99 | 1.00 | MGC |
| | 1 | 0.50 | 0.73 | 0.91 | 0.99 | 1.00 | MGC | 0.50 | 0.73 | 0.91 | 0.99 | 1.00 | MGC |
| | 2 | 0.50 | 0.72 | 0.90 | 0.99 | 1.00 | MGC | 0.50 | 0.72 | 0.90 | 0.99 | 1.00 | MGC |
| covariance | 0 | 0.50 | 0.73 | 0.90 | 0.99 | 1.00 | MGC | 0.50 | 0.74 | 0.90 | 1.00 | 1.00 | MGC |
| | 1 | 0.50 | 0.72 | 0.89 | 0.99 | 1.00 | MGC | 0.50 | 0.73 | 0.90 | 1.00 | 1.00 | MGC |
| | 2 | 0.50 | 0.73 | 0.91 | 0.99 | 1.00 | MGC | 0.50 | 0.74 | 0.91 | 1.00 | 1.00 | MGC |
| magnitude_dep | 0 | 0.50 | 0.73 | 0.88 | 0.99 | 1.00 | MGC | 0.50 | 0.73 | 0.89 | 0.99 | 1.00 | MGC |
| | 1 | 0.50 | 0.72 | 0.88 | 0.99 | 1.00 | MGC | 0.50 | 0.73 | 0.88 | 0.99 | 1.00 | MGC |
| | 2 | 0.50 | 0.73 | 0.89 | 0.99 | 1.00 | MGC | 0.50 | 0.73 | 0.89 | 0.99 | 1.00 | MGC |
| triple_interaction | 0 | 0.50 | 0.73 | 0.90 | 0.99 | 1.00 | MGC | 0.50 | 0.74 | 0.90 | 0.99 | 1.00 | MGC |
| | 1 | 0.50 | 0.73 | 0.89 | 0.99 | 1.00 | MGC | 0.50 | 0.73 | 0.89 | 0.99 | 1.00 | MGC |
| | 2 | 0.50 | 0.72 | 0.88 | 0.99 | 1.00 | MGC | 0.50 | 0.73 | 0.89 | 0.99 | 1.00 | MGC |
| parity4 | 0 | 0.50 | 0.73 | 0.89 | 0.99 | 1.00 | MGC | 0.50 | 0.73 | 0.89 | 1.00 | 1.00 | MGC |
| | 1 | 0.50 | 0.73 | 0.87 | 0.99 | 1.00 | MGC | 0.50 | 0.74 | 0.89 | 1.00 | 1.00 | MGC |
| | 2 | 0.50 | 0.71 | 0.88 | 0.99 | 1.00 | MGC | 0.50 | 0.73 | 0.89 | 0.99 | 1.00 | MGC |
| corr_pattern | 0 | 0.49 | 0.73 | 0.90 | 0.99 | 1.00 | MGC | 0.50 | 0.73 | 0.90 | 0.99 | 1.00 | MGC |
| | 1 | 0.52 | 0.76 | 0.91 | 0.99 | 1.00 | MGC | 0.50 | 0.76 | 0.90 | 0.99 | 1.00 | MGC |
| | 2 | 0.53 | 0.80 | 0.92 | 1.00 | 1.00 | MGC | 0.51 | 0.76 | 0.92 | 1.00 | 1.00 | MGC |
| missing_rate | 0 | 0.50 | 0.74 | 0.88 | 0.99 | 1.00 | MGC | 0.50 | 0.74 | 0.90 | 0.99 | 1.00 | MGC |
| | 1 | 0.50 | 0.72 | 0.88 | 0.99 | 1.00 | MGC | 0.50 | 0.73 | 0.90 | 0.99 | 1.00 | MGC |
| | 2 | 0.50 | 0.73 | 0.89 | 0.99 | 1.00 | MGC | 0.50 | 0.73 | 0.89 | 0.99 | 1.00 | MGC |
| missing_dep | 0 | 0.50 | 0.53 | 0.57 | 0.64 | 0.70 | MgC | 0.50 | 0.75 | 0.91 | 1.00 | 1.00 | MGC |
| | 1 | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | MgC | 0.50 | 0.75 | 0.91 | 1.00 | 1.00 | MGC |
| | 2 | 0.50 | 0.69 | 0.82 | 0.97 | 1.00 | MGC | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | MgC |
| anisotropy | 0 | 0.50 | 0.50 | 0.50 | 0.50 | 0.51 | MgC | 0.50 | 0.63 | 0.90 | 0.96 | 0.96 | MGC |
| | 1 | 0.50 | 0.50 | 0.52 | 0.55 | 0.56 | MgC | 0.50 | 0.64 | 0.91 | 0.95 | 0.95 | MGC |
| | 2 | 0.50 | 0.50 | 0.51 | 0.51 | 0.52 | MgC | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | MgC |
| cluster_sep | 0 | 0.50 | 0.50 | 0.82 | – | – | Mg- | 0.50 | 0.51 | 0.88 | – | – | Mg- |
| | 1 | 0.50 | 0.51 | 0.85 | – | – | Mg- | 0.50 | 0.50 | 0.88 | – | – | Mg- |
| | 2 | 0.50 | 0.51 | 0.85 | – | – | Mg- | 0.50 | 0.51 | 0.88 | – | – | Mg- |
| row_dependence | 0 | 0.50 | 0.51 | 0.56 | – | – | Mg- | 0.50 | 0.53 | 0.86 | – | – | Mg- |
| | 1 | 0.50 | 0.50 | 0.50 | – | – | Mg- | 0.50 | 0.54 | 0.84 | – | – | Mg- |
| | 2 | 0.50 | 0.51 | 0.61 | – | – | Mg- | 0.50 | 0.54 | 0.85 | – | – | Mg- |
| explicit | 0 | 0.50 | 0.83 | 1.00 | 0.97 | 0.96 | MGC | 0.50 | 1.00 | 1.00 | 1.00 | 1.00 | MgC |
| | 1 | 0.50 | 0.85 | 1.00 | 0.97 | 0.92 | MGC | 0.50 | 1.00 | 1.00 | 1.00 | 1.00 | MgC |
| | 2 | 0.50 | 0.87 | 1.00 | 0.99 | 0.96 | MGC | 0.50 | 1.00 | 1.00 | 1.00 | 1.00 | MgC |

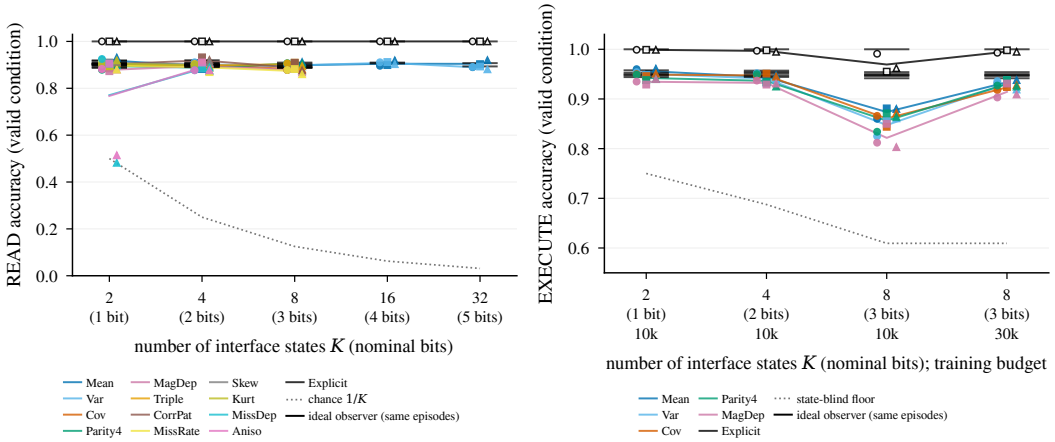
The full curves ($m \in \{0, 0.25, 0.5, 0.75, 1, 1.5, 2, 3\}$), bootstrap CIs, pair-correct and encoded-answer probabilities are in the CSV. cluster_sep and row_dependence leave their construction domain for $m > 1$ (skipped, recorded). Ideal-observer accuracies on the same episodes rise from 0.5 at $m = 0$ to $\approx 0.90$ at $m = 1$ and $\approx 1.0$ at $m = 3$ (per-cell values in the CSV).

### C.3 Multi-State Interface Capacity

We next increase the number of carrier states. For each carrier and state cardinality $K$, the carrier is recalibrated to approximately matched finite-context ideal-observer accuracy at $N = 128$.

For READ, every trained higher-$K$ condition remains close to its ideal observer. Eight statistical carriers support $K = 8$ in all three seeds, while mean and variance are additionally evaluated successfully at $K = 16$ and $K = 32$. Thus, READ demonstrates at least five nominal state bits for individual carriers in the tested setting.

For EXECUTE, we evaluate a narrower selected stress test at $K = 4$ and $K = 8$. At $K = 8$, covariance, mean, `parity4`, and variance reach $0.92$–$0.98$ task-oracle efficiency in every seed after the prospectively specified $30{,}000$-step extension. This establishes at least three nominal executable control bits (Figure 9).




(a) (b)

Figure 9: **Multi-state statistical interfaces. (a)** READ remains close to the ideal observer throughout the trained higher-$K$ conditions. **(b)** Statistical states select among up to eight computations under EXECUTE.

These experiments provide lower bounds, not measurements of maximum capacity. Several carrier–$K$ combinations are not trained because the carrier itself cannot reach the predefined finite-context calibration target within its admissible strength range.

Table 9: Multi-state interface summary per carrier: the $K$ values trained, the largest $K$ at which every seed reaches $\geq 0.90$ of the ideal observer (READ; EXECUTE at 10k and at the 30k re-run), the corresponding nominal state bits, and the construction ceiling (the largest $K$ whose calibration to $\mathrm{Acc}^*(128) = 0.90$ converged).

| carrier | audit | READ $K$ trained | largest READ $K$ (3/3) | bits | EXEC $K$ trained | largest EXEC $K$ 10k | same 30k | bits | ceiling $K$ |
|---|---|---|---|---|---|---|---|---|---|
| mean | CLEAN | 2, 4, 8, 16, 32 | 32 | 5 | 2, 4, 8 | 4 | 8 | 3 | 32 (calibration converged) |
| variance | CLEAN | 2, 4, 8, 16, 32 | 32 | 5 | 2, 4, 8 | 4 | 8 | 3 | 32 (calibration converged) |
| covariance | CLEAN | 2, 4, 8 | 8 | 3 | 2, 4, 8 | 4 | 8 | 3 | 8 (calibration converged) |
| parity4 | CLEAN | 2, 4, 8 | 8 | 3 | 2, 4, 8 | 4 | 8 | 3 | 8 (calibration converged) |
| magnitude_dep | CLEAN | 2, 4, 8 | 8 | 3 | 2, 4, 8 | 4 | – | 2 | 8 (calibration converged) |
| triple_interaction | CLEAN | 2, 4, 8 | 8 | 3 | – | – | – | – | 8 (calibration converged) |
| corr_pattern | CLEAN | 2, 4, 8 | 8 | 3 | – | – | – | – | 8 (calibration converged) |
| missing_rate | CLEAN | 2, 4, 8 | 8 | 3 | – | – | – | – | 8 (calibration converged) |
| skewness | CLEAN | 2, 4 | 4 | 2 | – | – | – | – | 4 (calibration converged) |
| kurtosis | CLEAN | 2, 4 | 4 | 2 | – | – | – | – | 4 (calibration converged) |
| missing_dep | CLEAN | 2, 4 | 4 | 2 | – | – | – | – | 4 (calibration converged) |
| anisotropy | CLEAN | 2, 4 | 4 | 2 | – | – | – | – | 4 (calibration converged) |
| explicit | NO CLAIMS (control) | 2, 4, 8, 16, 32 | 32 | 5 | 2, 4, 8 | 4 | 8 | 3 | 32 (control; not calibrated) |

Nominal bits $= \log_2 K$ of reliably distinguished states – a demonstrated lower bound at matched finite-context evidence, not a channel capacity. Untrained $K$ are cells whose bisection ended at the bracket end below the target (Appendix B.4): the construction cannot present that many states at $\mathrm{Acc}^*(128) = 0.90$ within its admissible range; no endpoint strength was trained. EXECUTE was run at $K > 2$ for five carriers chosen after the atlas as the strong, $K = 8$-calibratable set (plus the explicit control); its largest $K$ is bounded by the experiment ($K \leq 8$), not by a failure.

### C.4 Prospectively Sampled Structural Interfaces

The hand-designed atlas isolates interpretable statistical properties but could favor structures chosen because they appear convenient. We therefore define a small structural-equation DSL and sample the interface family before observing any PFN learning outcome.

Each interface is a four-variable structural model

$$x_j = \mu_j(x_{<j}) + \exp(\ell_j(x_{<j}))\,\epsilon_j, \qquad \epsilon_j \sim \mathcal{N}(0,1), \tag{11}$$

with six primitive operations:

$$\begin{aligned}
\mathrm{SHIFT}(j): &\quad \mu_j \mathrel{+}= c,\\
\mathrm{SCALE}(j): &\quad \ell_j \mathrel{+}= c,\\
\mathrm{LIN}(i\to j): &\quad \mu_j \mathrel{+}= c x_i,\\
\mathrm{QUAD}(i\to j): &\quad \mu_j \mathrel{+}= c(x_i^2 - 1),\\
\mathrm{PROD}(i,k\to j): &\quad \mu_j \mathrel{+}= c x_i x_k,\\
\mathrm{HETERO}(i\to j): &\quad \ell_j \mathrel{+}= c\,\mathrm{sign}(x_i).
\end{aligned} \tag{12}$$

Parents precede their target, yielding an acyclic structural model. Each program contains one to three operations, with a non-empty subset designated state-bearing. For those operations,

$$c = a\,\sigma_z\,\alpha, \qquad \sigma_0 = +1, \quad \sigma_1 = -1, \tag{13}$$

where the sampled coefficient $a$ is frozen with the program and $\alpha$ is the interface-strength parameter.

Exactly 48 programs are sampled from the DSL with a fixed generator seed. No larger candidate pool is generated, and no program is replaced, removed, or redrawn based on PFN outcomes. Their strengths are calibrated using only the generator and ideal observer before any PFN is trained. Figure 10 shows representative sampled interfaces.

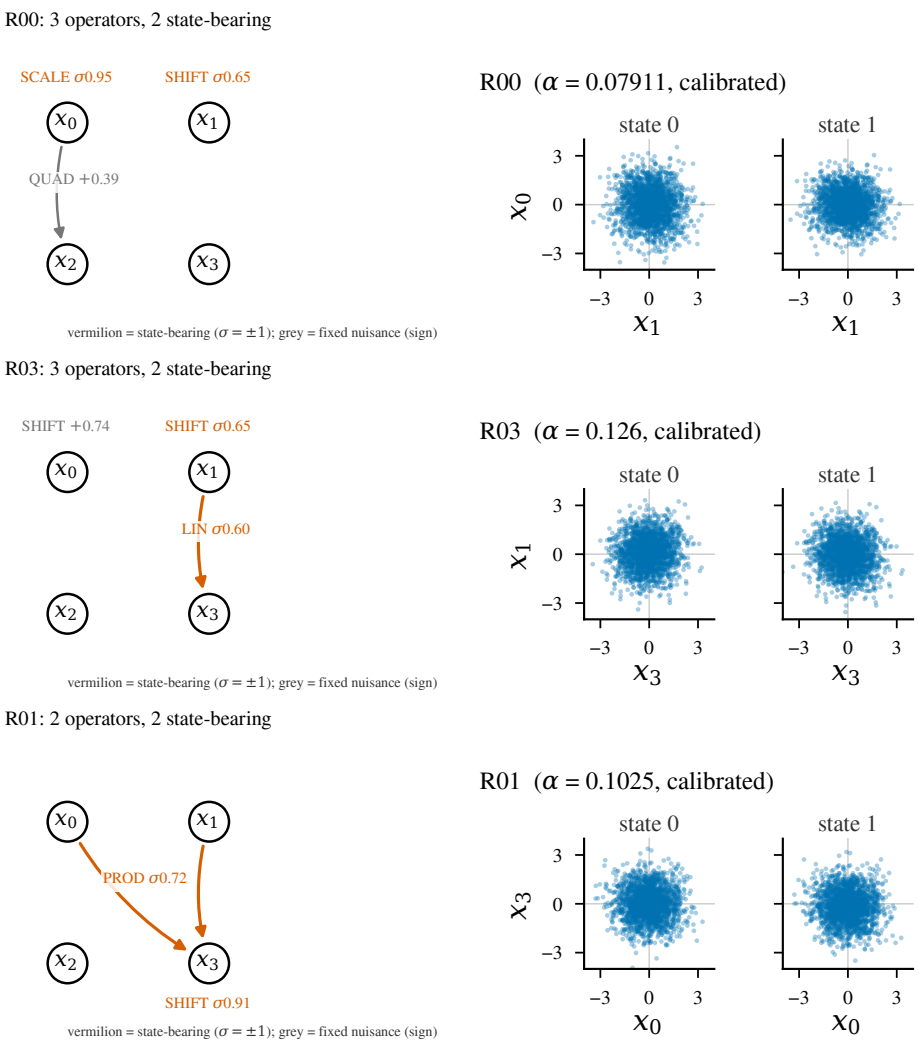



Figure 10: **Examples from the prospectively sampled structural-interface DSL.** Left: frozen structural programs. Right: samples from their two states at the calibrated experimental strength. The subtle differences are intentional because calibration targets discrimination from an entire $N = 128$ context rather than visually obvious single-row separation.

### C.5 Acquisition Across 48 Matched Interfaces

All 48 sampled interfaces are calibrated independently to approximately matched finite-context ideal-observer accuracy before training and are then evaluated with three PFN seeds.

We define acquisition cost $L_C$ as the first evaluated checkpoint at which

$$\frac{\mathrm{Acc}_{\mathrm{model}} - 0.5}{\mathrm{Acc}^{*} - 0.5} \geq 0.9. \tag{14}$$

The evaluated checkpoints are 250, 500, 1,000, 2,000, 5,000, and 10,000 optimization steps.

All 144/144 runs acquire within the training budget. Across runs, first crossing ranges from 250 to 5,000 steps. Across carrier means, $L_C$ ranges from 250 to 3,000 steps, a twelve-fold difference despite matched finite-context decodability (Figure 3).

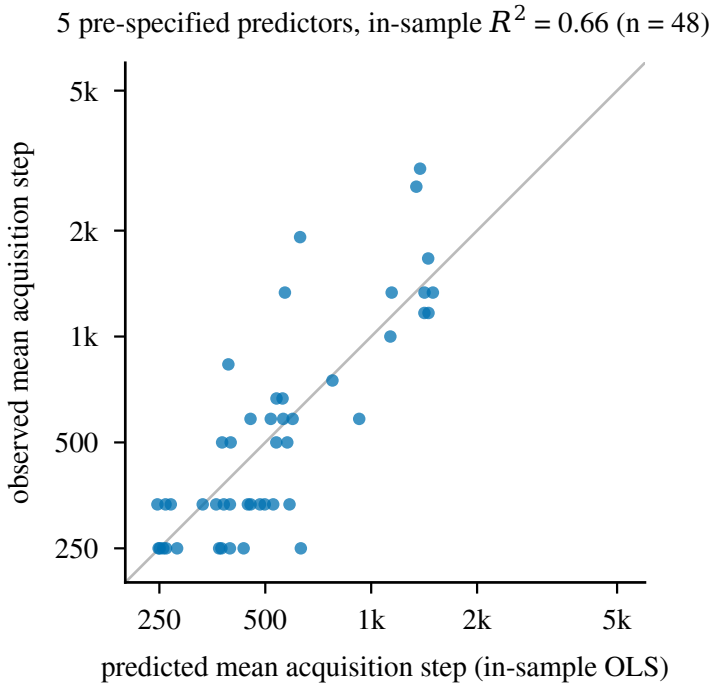



Figure 11: **Predicting acquisition cost from pre-specified interface descriptors.** Observed versus fitted log mean acquisition cost for an ordinary-least-squares model using five interface descriptors fixed before training. The in-sample model explains $R^2 = 0.664$ of the variance.

Acquisition difficulty is systematically associated with interface properties known before training. Maximum interaction order correlates with log mean acquisition cost at Spearman $\rho = 0.69$, while the presence of nonlinear state-bearing operators has $\rho = 0.60$. Five pre-specified descriptors jointly explain $66.4\%$ of the in-sample variation in log mean acquisition cost (Figure 11). These analyses characterize variation within the frozen DSL distribution; they do not imply universal learnability of arbitrary statistical properties.

As a descriptive robustness check performed after the frozen analysis, leave-one-out prediction retains $R^2 = 0.547$. The reported Spearman coefficients use standard tie-corrected midranks.

Table 10: Acquisition of the 48 frozen random structural interfaces (READ, $K = 2$, $N = 128$, 10 000 steps, seeds 0–2). $L_C$ = first evaluated checkpoint (250, 500, 1 000, 2 000, 5 000, 10 000) at which the model's accuracy reaches $\geq 0.9$ of the above-chance ideal-observer accuracy on the same evaluation episodes (censored at 10 000 if never). The reported acquisition quantity is the mean of $L_C$ over the three seeds; The five predictors used for the Spearman and ordinary-least-squares analyses were fixed before training.

| quantity | value | status / source |
|---|---|---|
| interfaces / seeds / runs | 48 / 3 / 144 | prospectively sampled interface family (generator seed 2026) |
| runs completed at 10 000 steps | 144 / 144 | run records |
| runs acquired (efficiency $\geq 0.90$ at some checkpoint) | 144 / 144; censored at budget: 0 | pre-specified $L_C$ definition |
| interfaces acquired in 3 / 3 seeds | 48 / 48 | raw records |
| final efficiency at 10k (all runs) | min 0.964, median 0.998 | raw records |
| $L_C$ per run (grid 250 / 500 / 1k / 2k / 5k / 10k) | 250: 71; 500: 34; 1000: 23; 2000: 13; 5000: 3 | raw records; min 250, median 250, max 5 000 |
| mean $L_C$ per interface | min 250, median 417, max 3000 | mean over three seeds |
| median $L_C$ per interface | 250: 27; 500: 11; 1000: 7; 2000: 3 | descriptive |
| interfaces whose three seeds acquire at the same checkpoint | 12 / 48 | descriptive |
| largest seed spread (max−min) | R32: 4750; R39: 4000; R41: 3000 | descriptive |
| Spearman $\rho$ (max_order vs. log mean $L_C$), $n = 48$ | +0.69 | pre-specified predictor; standard tie-corrected Spearman |
| Spearman $\rho$ (has_nonlinear vs. log mean $L_C$), $n = 48$ | +0.60 | pre-specified predictor; standard tie-corrected Spearman |
| Spearman $\rho$ (marginal_changes vs. log mean $L_C$), $n = 48$ | -0.45 | pre-specified predictor; standard tie-corrected Spearman |
| Spearman $\rho$ (per_row_kl vs. log mean $L_C$), $n = 48$ | -0.26 | pre-specified predictor; standard tie-corrected Spearman |
| Spearman $\rho$ (probe_marginal vs. log mean $L_C$), $n = 48$ | -0.49 | pre-specified predictor; standard tie-corrected Spearman |
| OLS on log mean $L_C$ with the 5 predeclared predictors | in-sample $R^2$ = 0.664; leave-one-out $R^2$ = 0.547 | in-sample: primary analysis; LOO: descriptive robustness check |

The frozen implementation ranks ties by position (no midranks); Spearman coefficients use standard tie-corrected midranks. The OLS uses log per-row KL and no standardisation; predictions are in-sample (leave-one-out $R^2$ added as a descriptive check). Interfaces are independent draws of the generator but four pairs share an operator structure.

### C.6 Coexistence and Cross-Talk

To test whether several interfaces can share one context, we use DUAL-READ. Two carrier blocks independently encode two binary states in the same context, and query rows separately address the two blocks.

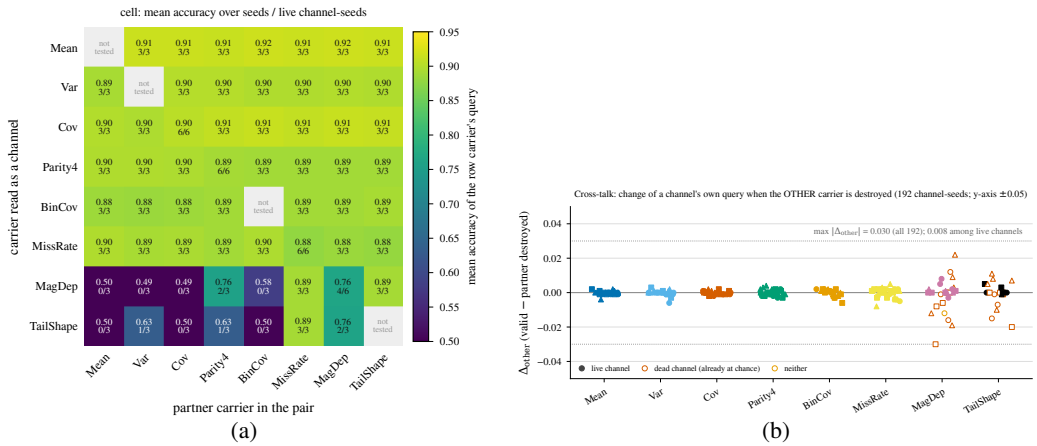
The primary pair matrix contains eight carriers: `mean`, `variance`, `covariance`, `parity4`, `magnitude_dep`, `missing_rate`, `binary_cov`, and `tail_shape`. We train all 28 heterogeneous pairs plus four homogeneous controls, each with three seeds, for 96 trained models and 192 channel–seed measurements.

Twenty of the 32 tested pairs retain both channels in all three seeds: 17 of 28 heterogeneous pairs and three of four homogeneous controls. Failures are confined to channels using `magnitude_dep` or `tail_shape`.

Selective interventions separate dependence on a channel's own carrier from cross-channel effects:

$$\Delta^{(i)}_{\text{own}} = A^{(i)}_{\text{valid}} - A^{(i)}_{\text{broken}_i}, \qquad \Delta^{(i)}_{\text{other}} = A^{(i)}_{\text{valid}} - A^{(i)}_{\text{broken}_j}, \quad j \neq i. \tag{15}$$

Across all 192 channel–seed measurements, $\max |\Delta_{\text{other}}| = 0.030$. Restricting to the 163 functional channels reduces this maximum to 0.008 (Figure 12).




(a) (b)

Figure 12: **Coexistence of statistical interfaces. (a)** Complete tested carrier-pair matrix. Failures are localized to particular statistical structures. **(b)** Selective interventions reveal little measured cross-talk once both interfaces are functional.

The coexistence boundary is therefore predominantly an acquisition boundary: particular carriers can fail to become functional when trained alongside another interface, but already functional channels behave nearly as they do in isolation.

Table 11: Conditions, definitions and denominators of the coexistence experiment, with an earlier three-pair instance and an architecture-varied replication. PRE-SPECIFIED = fixed in the frozen specification before training; FROZEN DERIVED = computed here from the frozen run records under a stated rule; POST-HOC = rule introduced after the results; NEW DESCRIPTIVE = quantity introduced in this appendix.

| item | definition / value | status |
|---|---|---|
| task | dual read: two carrier blocks (4 columns each) in the same context rows, independent binary states; two query rows, the operand column addresses block 0 or 1; target = state of the addressed block | PRE-SPECIFIED |
| model / training | PFN 64/3/4/128, one feature group, no positional information; AdamW $10^{-3}$, cosine, 5 % warm-up, batch 64, 10 000 steps; parameter count 107 714 (107 394 for a single READ interface: one extra operand column) | frozen run configuration |
| calibration | each carrier at its frozen single-interface $K = 2$ strength ($\mathrm{Acc}^*(128) \approx 0.90$, Appendix B.4); no re-calibration for the pair | PRE-SPECIFIED |
| conditions (paired episodes) | valid; broken carrier 0 / broken carrier 1 (one carrier re-rendered with an independently drawn state, the other untouched); broken (both); permuted (row shuffle; equal to valid in all 96 runs) | PRE-SPECIFIED (episode conditions, Appendix B.2) |
| per-channel quantities | $\mathrm{valid}_i$ = accuracy of channel $i$'s query; $\Delta_{\mathrm{own}}$ = $\mathrm{valid}_i$ − accuracy under broken carrier $i$; $\Delta_{\mathrm{other}}$ = $\mathrm{valid}_i$ − accuracy under broken carrier $j$ ($j \neq i$); 1 000 episodes, seed 90000+seed, final checkpoint | PRE-SPECIFIED (spec §1) |
| dead channel | valid within 0.05 of chance AND $\Delta_{\mathrm{own}} \leq 0.05$ (the pre-specified rule uses OR; both rules give the same 28 channel-seeds here) | PRE-SPECIFIED |
| live channel | valid $\geq 0.85$ AND $\Delta_{\mathrm{own}} \geq 0.32$; a pair-seed has "both live" if both channels are live | POST-HOC |
| pre-specified hypothesis | the channel that dies is the carrier with the larger single-interface $L_C$ | PRE-SPECIFIED |
| pool / design | 8 carriers (6 hand-designed references plus `binary_cov` and `tail_shape`, selected by the pre-specified novelty rule); all 28 heterogeneous pairs + 4 homogeneous pairs; 3 seeds; 96 runs, 96 completed, 0 invalid | PRE-SPECIFIED design; FROZEN records |
| channel-seeds | 192 = 32 pairs × 2 channels × 3 seeds: 163 live, 28 dead, 1 neither | FROZEN DERIVED |
| pairs with both channels live in 3/3 seeds | 20/32 (heterogeneous 17/28, homogeneous 3/4) | FROZEN DERIVED |
| dead channels | 28 channel-seeds in 12 pairs; carriers: MagDep, TailShape; every dead channel's accuracy stays ≤ 0.543 at every checkpoint | FROZEN DERIVED |
| cross-talk | max $\lvert\Delta_{\mathrm{other}}\rvert$ = 0.030 over all 192 channel-seeds; 0.008 over the 163 live channel-seeds; none exceeds the status-script cross-talk flag of 0.10 | FROZEN DERIVED (0.10 threshold fixed before training) |
| live channel vs. single interface | in-pair minus single-interface accuracy (same carrier and seed; single-interface read screens): mean −0.001, range [−0.042, +0.027]; efficiency against the single-interface ideal observer: min 0.899, median 0.987 | NEW DESCRIPTIVE (different evaluation episodes) |
| pre-specified hypothesis | 26 heterogeneous dead channel-seeds: in all of them the dead carrier has the larger mean single-interface $L_C$ (necessary); `missing_rate` (largest $L_C$, 6 667) never dies (not sufficient) | PRE-SPECIFIED test; FROZEN DERIVED |
| earlier three-pair instance | 3 pairs × 3 seeds; both channels live 3/3 in 2/3 pairs; `magnitude_dep` dead in all 3 seeds with `variance`; max $\lvert\Delta_{\mathrm{other}}\rvert$ = 0.022 | FROZEN DERIVED (per-channel criterion POST-HOC) |
| architecture variation | `covariance+missing_rate` under small / wide / deep: 18/18 channel-seeds live; max $\lvert\Delta_{\mathrm{other}}\rvert$ = 0.003 | FROZEN DERIVED |

Single-interface baselines: the single-carrier read screens covering all eight carriers, at the same carrier strength, $N$, $K$, budget and seed; their evaluation episodes differ from the pair runs' (different episode layout), so the comparison is descriptive. The pooled two-query accuracy and the 'oracle efficiency' recorded in the pair runs' evaluation files are not used: the recorded oracle is the ideal observer of carrier A alone.

### C.7 Robustness Across PFN Scale

The preceding experiments use a compact PFN with approximately $1.08 \times 10^5$ parameters. We additionally test whether statistical-interface acquisition survives increased model capacity within the same PFN architecture family.

Five independently trained scales span $107{,}908$ to $48{,}382{,}468$ parameters, a factor of $448.4$. Only model width, depth, number of attention heads, and feed-forward width are changed. Larger models require smaller peak learning rates for stable optimization, so the scaling claim concerns robustness under a scale-specific stability-selected training recipe rather than a fixed learning rate (Table 12).

Table 12: **PFN scaling ladder.** The scaling study spans five independently trained sizes and ends at 48.4M parameters.

| Model | Parameters | × smallest | Layers | Peak lr | stable/acquired |
|---|---|---|---|---|---|
| Default (compact) | 107,908 | 1.0 | 3 | $1.0 \times 10^{-3}$ | 36/36 |
| Wider | 559,876 | 5.19 | 4 | $1.0 \times 10^{-3}$ | 36/36 |
| Medium | 3,282,436 | 30.4 | 6 | $2.0 \times 10^{-4}$ | 36/36 |
| Large | 17,312,772 | 160.4 | 8 | $2.0 \times 10^{-4}$ | 35/36 |
| Largest | 48,382,468 | 448.4 | 10 | $1.5 \times 10^{-4}$ | 36/36 |

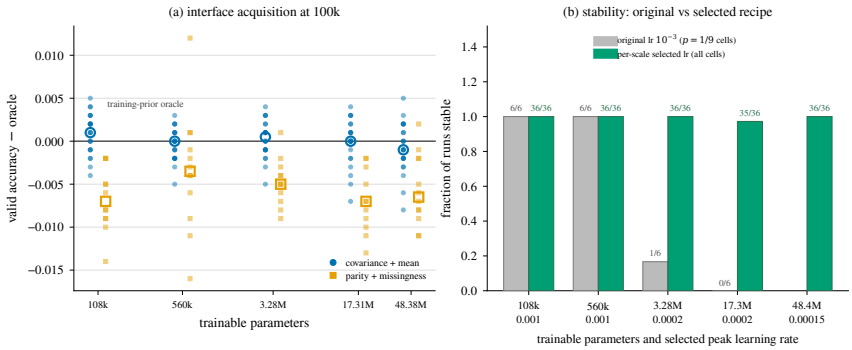



Figure 13: **Acquisition and optimization stability across PFN scale.** Larger models continue to acquire the statistical interface, but optimization becomes unstable under learning rates that remain viable at smaller scales. The scaling evidence covers all five model sizes.

This experiment scales the READ-type interface testbed rather than repeating the complete WRITE/EXECUTE/coexistence stack. Figure 13 shows that every size acquires the interface under its selected recipe, whereas the original learning rate destabilizes the two intermediate sizes. It therefore supports robustness of acquisition across this PFN family, not architecture-invariant statistical control in general.

We additionally repeat the off-training evidence-strength intervention used in Appendix D.2. Among stable runs, the training-distribution posterior is closer to the frozen model in every evaluated off-training cell: $150/150$ at 107.9k, 559.9k, and 3.28M parameters, $145/145$ at 17.3M, and $150/150$ at 48.4M parameters. The smaller 17.3M denominator reflects one run failing the frozen stability criterion (Figure 14).

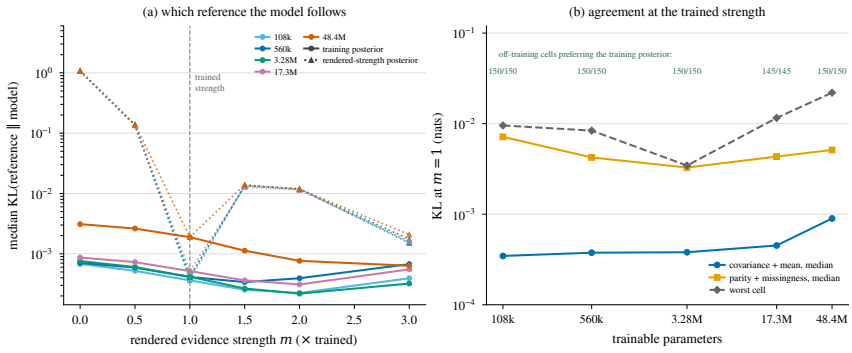



Figure 14: **Learned evidence semantics across PFN scale.** Left: median KL divergence to the training-distribution posterior and to the posterior implied by rendered evidence strength under off-training interventions. Right: agreement at the trained strength, where the two references coincide. Counts report off-training cells favoring the training-distribution reference.

The result establishes qualitative persistence rather than numerical invariance. At the trained carrier strength, quantitative agreement varies across scale and the largest architecture exhibits larger worst-case deviations. Likewise, the scaling experiment does not test EXECUTE, interface capacity, or multi-channel coexistence. It therefore shows that statistical readout and its learned training-distribution-dependent evidence semantics persist across a $448\times$ increase in PFN size under scale-specific stability-selected optimization.

### C.8 Boundaries and Failure Modes

The controlled experiments expose several distinct failure modes. They should not be collapsed into a single statement that an interface "works" or "fails." Table 13 separates them.

Table 13: **Distinct boundaries observed in the controlled testbed.**

| Boundary | Example | Interpretation |
|---|---|---|
| Architectural access | `row_order`: identical row multiset in a different order | State information removed by permutation-equivariant input representation; not a learning failure. |
| Acquisition under the tested regime | `modality`: symmetric mixtures with matched first four moments, ideal-observer decodable but not reliably acquired | Statistical availability does not guarantee acquisition at a fixed architecture, context size, seed, and budget. |
| Task-specific acquisition | `anisotropy`: strong READ in two seeds, no reliable EXECUTE | Learning to recover a state and learning to use it computationally are distinct problems. |
| Intervention criterion | `corr_pattern`: rewrite succeeds, nominal null retains information | Failure of a strict null construction need not imply failure of causal control. |
| Simultaneous acquisition | `magnitude_dep`, `tail_shape` fail in selected pairs | Interfaces may work in isolation yet fail to be jointly acquired; successful channels show little cross-talk. |

Statistical information must first be available to the model, then acquired by the learner, and finally used by the downstream task. Failures can occur at each stage. Accordingly, statistical decodability is necessary for the interfaces studied here but is not sufficient for their acquisition or execution under a particular training regime.

## D Extended Results: Interface Semantics

### D.1 Operational Semantics Can Be Rebound

The counterfactual experiments in Appendix C.2 rewrite a physical carrier state while keeping its learned meaning fixed. Here we perform the complementary intervention: we keep the physical carrier fixed and change the meaning assigned to it during training.

For a binary carrier state $s$, paired models are trained with either the forward mapping,

$$s \mapsto \mathrm{op}_s,$$

or the reversed mapping,

$$s \mapsto \mathrm{op}_{1-s}\,.$$

Architecture, carrier construction, calibrated strength, state prior, operand distribution, and optimization are otherwise identical. After training, forward- and reverse-trained models are evaluated on bit-identical episodes.

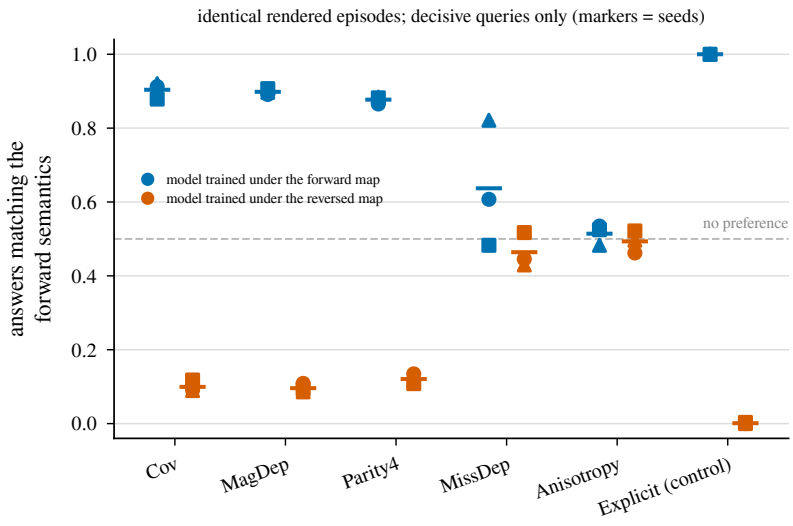



Figure 15: **Training can rebind the operational meaning of a statistical state.** Forward- and reverse-trained models receive identical physical evaluation inputs but have learned opposite state-to-operation mappings. For covariance, magnitude dependence, fourth-order parity, and the explicit control, changing only the training mapping reverses which computation is selected. Carriers that fail to acquire EXECUTE reliably do not establish robust rebinding.

For covariance, magnitude dependence, and fourth-order parity, all model seeds that acquire the executable interface follow the mapping assigned during training (Figure 15). Across these three statistical carriers, the fraction of decisive queries following the model's own mapping is $0.865$–$0.921$, compared with $0.079$–$0.135$ for the opposite mapping. Own-minus-opposite differences range from $+0.731$ to $+0.841$, with confidence intervals excluding zero in all $18/18$ statistical model-seeds.

This reversal cannot be explained by simply complementing the model output. The two binary operations agree on half of operand space, so output inversion would achieve only approximately $0.5$ task accuracy. Models that acquire their mapping instead reach $0.929$–$1.000$ raw EXECUTE accuracy.

The negative cases are also informative. Missing dependence and anisotropy do not establish robust rebinding at the tested budget, but these models also fail to reliably acquire the corresponding EXECUTE interface. Thus, readability alone is insufficient for semantic reassignment: the carrier must first become an executable interface.

Together with the inference-time rewriting experiment, this separates physical carrier state from learned semantics. A carrier state can be changed while its learned meaning is held fixed, and its learned meaning can be reversed while the physical input is held fixed.

### D.2 Training-Distribution Semantics of Statistical Evidence

We next test the quantitative meaning assigned to statistical evidence. Each model is trained under a fixed prior $p = P_{\text{train}}(z^\star)$ over a distinguished joint state and at one native carrier strength $s$. After training, model parameters are frozen and only the rendered evidence strength is varied:

$$m \in \{0, 0.5, 1, 1.5, 2, 3\},$$

where $m = 1$ is the strength encountered during training.

For every rendered context we compute two exact posterior references. $\text{Bayes}_{\text{true}}$ uses the likelihood associated with the actual rendered strength $ms$:

$$p_{\text{true}}(z \mid C_m) \propto p_{\text{train}}(z)\, p(C_m \mid z; ms), \tag{16}$$

whereas $\text{Bayes}_{\text{train}}$ scores the same rendered rows using the likelihood associated with the native training strength:

$$p_{\text{train-lik}}(z \mid C_m) \propto p_{\text{train}}(z)\, p(C_m \mid z; s). \tag{17}$$

The two references coincide at $m = 1$ and diverge when evidence is rendered weaker or stronger than during training. Neither reference depends on the model output.

For the primary covariance-plus-mean interface, the experiment spans seven nonzero training priors from $3 \times 10^{-4}$ to $0.25$, three model seeds, and five off-training evidence strengths. $\text{Bayes}_{\text{train}}$ is closer to the model in every one of the resulting $105/105$ comparisons. The median absolute deviation is $0.0069$ from $\text{Bayes}_{\text{train}}$ and $0.0698$ from $\text{Bayes}_{\text{true}}$; median four-state KL divergence is $0.0059$ versus $0.2161$ nats.

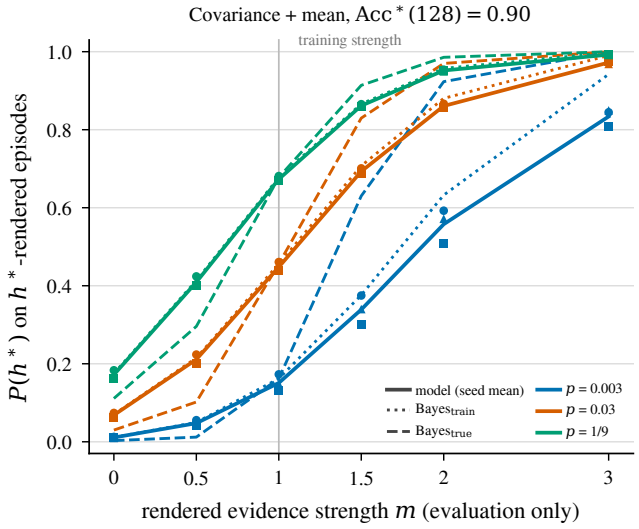



Figure 16: **Training-distribution semantics of statistical evidence.** Models are trained at native strength $m = 1$ and evaluated with frozen weights while evidence strength is varied. Solid curves show model posterior mass, dotted curves $\text{Bayes}_{\text{train}}$, and dashed curves $\text{Bayes}_{\text{true}}$. Away from $m = 1$, behavior follows the likelihood semantics associated with the training distribution more closely than the physical intervention strength.

The effect is visible both below and above the training strength (Figure 16). At $m = 0$, the physical carrier contains no state-dependent signal, so $\text{Bayes}_{\text{true}}$ reduces to the training prior. Nevertheless, the model remains substantially closer to $\text{Bayes}_{\text{train}}$. At $m = 1.5$ and $m = 2$, the same ordering persists when the rendered signal is stronger than encountered during training. At $m = 3$, both references approach saturation and quantitative agreement weakens, but $\text{Bayes}_{\text{train}}$ remains the closer reference in every tested cell.

At native strength, where the references coincide, agreement with the exact training posterior is correspondingly tight: covariance plus mean yields prior-weighted pooled four-state KL divergence of $0.0020$–$0.0027$ nats.

A second independently constructed family, fourth-order parity plus missing rate, reproduces the direction of the result. $\text{Bayes}_{\text{train}}$ is closer in all $45/45$ off-training cells, with median absolute deviations $0.0295$ and $0.0750$ from $\text{Bayes}_{\text{train}}$ and $\text{Bayes}_{\text{true}}$, respectively. Its quantitative agreement is weaker than for covariance plus mean, so we use it as a qualitative replication rather than as evidence for the same near-exact alignment. Table 14 records both carrier families.

Table 14: **Off-training evidence is interpreted according to learned likelihood semantics.** Values aggregate the frozen off-training evidence-strength comparisons.

| Carrier family | cells favoring $\text{Bayes}_{\text{train}}$ | median error to $\text{Bayes}_{\text{train}}$ | median error to $\text{Bayes}_{\text{true}}$ |
|---|---|---|---|
| Covariance + mean | $105/105$ | $0.0069$ | $0.0698$ |
| Parity4 + missing rate | $45/45$ | $0.0295$ | $0.0750$ |

These results are behavioral. They show that, within the controlled training families studied here, frozen PFN predictions quantitatively reflect the prior and likelihood semantics of their training distribution. They do not establish that the network explicitly represents priors, likelihoods, or posterior distributions internally, nor that it implements a particular Bayesian inference algorithm.

### D.3 Training Support and Out-of-Support Composition

We test whether individually readable statistical variables can be recombined into joint states assigned exactly zero probability during training. The manipulation concerns joint support, not marginal familiarity: every individual factor value remains represented in the training distribution.

For the primary two-factor family, each factor takes $K$ values and selected pairs are removed from the training generator. For example, at $K = 4$ we exclude

$$\{(0, 0), (1, 3), (2, 2), (3, 1)\}, \tag{18}$$

equivalently the pairs satisfying

$$(a + b) \bmod 4 = 0. \tag{19}$$

The construction is extended to $K \in \{2, 3, 4, 6\}$ and to three- and four-factor settings. For three $K = 3$ factors, tuples satisfying $(a + b + c) \bmod 3 = 0$ are excluded while every individual value and pairwise combination remains represented. For four binary factors, one parity class is excluded while all lower-order marginals remain represented. Zero support is imposed directly by the generator rather than arising from finite sampling.

**Factorized and support-constrained references.** We compare the model with two model-free decoders. The *factorized oracle* independently decodes each carrier from its exact likelihood and permits arbitrary recombination. The *support-projection oracle* uses the same carrier evidence but restricts the joint prediction to states with nonzero training probability. Consequently, its joint accuracy on an excluded tuple is exactly zero. The factorized oracle performs essentially identically on supported and held-out tuples, confirming that the excluded combinations are not themselves harder to decode from the individual carrier blocks.

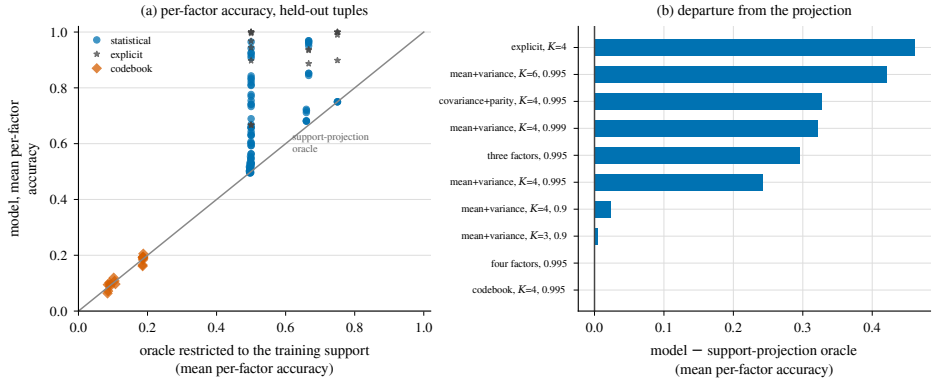



Figure 17: **Predictions on zero-support combinations reflect the joint training support.** At ordinary carrier strength, factor-level behavior on held-out combinations is close to a decoder that uses the available statistical evidence but projects its prediction onto the support of the training distribution. The effect is graded rather than absolute.

At the standard calibration of approximately $\mathrm{Acc}^* = 0.90$, the statistical models recover little of the held-out joint state despite retaining information about its components (Figure 17). Across representative two-factor conditions, held-out joint accuracy is $0.015$–$0.136$ against factorized-oracle accuracies of approximately $0.80$. The corresponding three-factor condition reaches $0.194$–$0.224$ against an oracle of $0.704$–$0.716$. When the same joint state is restored to training support, accuracy rises to $0.790$–$0.821$ in a matched control, compared with $0.0005$–$0.042$ when that state has zero training probability.

**The support boundary is graded.** With nearly unambiguous individual carriers, substantial residual out-of-support recombination can occur. Representative $K = 4$ conditions reach held-out accuracy of $0.467$–$0.686$, $K = 6$ conditions $0.816$–$0.884$, and a high-evidence three-factor condition $0.852$–$0.906$. These values remain below their factorized references, and only $3/78$ statistical run-seeds satisfy the pre-specified recombination criterion. Recovery also depends on support geometry: increasing the number of trained partners for each factor value increases recombination, whereas a four-factor parity restriction produces almost none despite highly informative individual carriers (Figure 18).

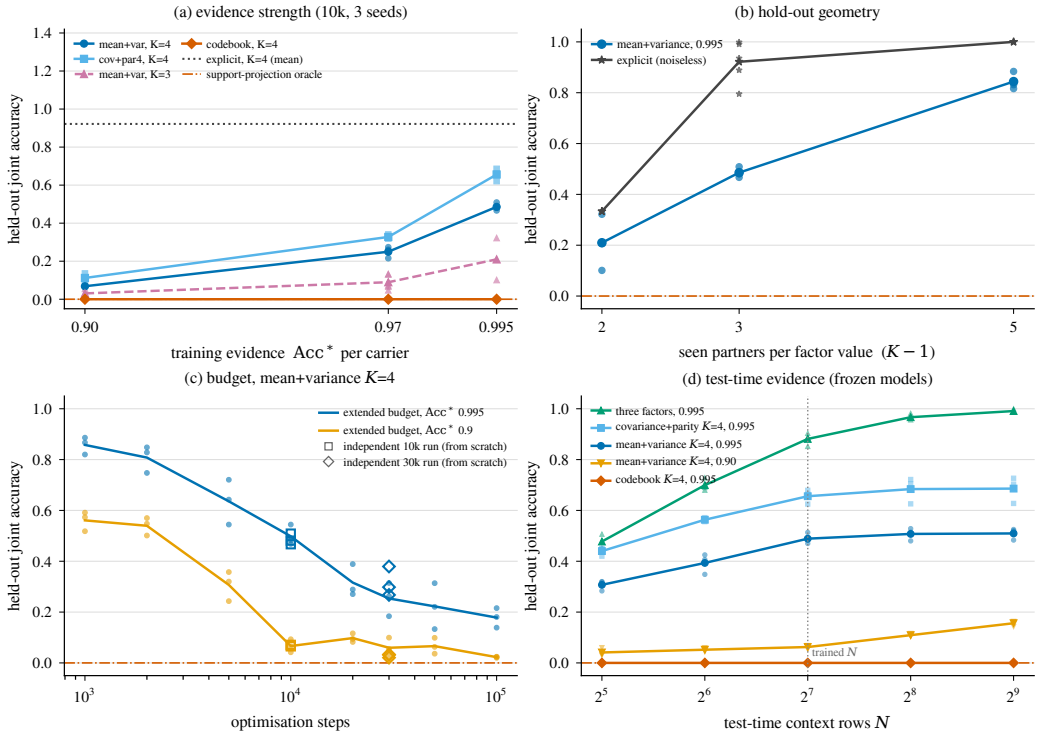



Figure 18: **The training-support boundary depends on evidence, support geometry, optimization, and test-time context.** The panels summarize residual recombination under stronger carrier evidence, different support geometries, continued optimization, and changes in context length. Zero training support constrains but does not universally prohibit production of an unseen joint state.

**Residual recombination decreases with optimization.** In independent from-scratch 10k and 30k runs, held-out joint accuracy decreases in all 18 matched cell-seeds while supported-state performance is maintained or improves. For the high-evidence three-factor condition, held-out accuracy falls from $0.852$–$0.906$ to $0.538$–$0.561$, while supported-tuple accuracy rises from $0.986$–$0.990$ to $0.990$–$0.993$.

A separately specified 100k trajectory provides a stronger control because all checkpoints share one optimization trajectory. For the primary high-evidence $K = 4$ condition, the three seeds move from

$$0.544,\ 0.482,\ 0.468$$

held-out accuracy at 10k steps to

$$0.314,\ 0.184,\ 0.263$$

at 30k and

$$0.139,\ 0.181,\ 0.216$$

at 100k, while supported accuracy rises from $0.983$–$0.988$ to $0.991$–$0.995$. The remaining held-out accuracy at 100k is still well above zero, so this is evidence for a directional optimization effect rather than convergence to the support-projection oracle.

**Support is not a representation-independent prohibition.** Increasing context length at the familiar carrier strength can overcome much of the boundary: in the high-evidence three-factor model, increasing test context from $N = 32$ to $N = 512$ raises held-out joint accuracy from $0.45$–$0.51$ to $0.99$–$1.00$. In contrast, simply rendering each row more strongly than encountered during training does not have the same effect and can degrade performance.

Representation also matters. A non-factorized codebook that represents the whole tuple by one statistical state yields exactly zero held-out accuracy in all tested runs, whereas noiseless explicit factor representations can recover several zero-support combinations. Thus, zero prior support strongly shapes behavior in the statistical interface setting but is not an absolute neural-network capacity barrier.

**Restricted support induces cross-channel dependence.** Under full joint support, interventions on one carrier have negligible effects on the readout of another, as in Appendix C.6. When the training distribution constrains the joint states, this changes. Representative off-diagonal intervention effects rise to approximately $0.15$–$0.21$, and in the four-factor parity condition the model reconstructs one state from the remaining carriers rather than relying on its own carrier. The meaning of one statistical interface can therefore depend on the joint training distribution of the others.

**Held-out tuples are heterogeneous.** Residual recombination is not uniform across excluded states. Tuples in which all factors take the same value are easier than mixed held-out tuples in 21 of 29 evaluated cells. For example, in a high-evidence statistical $K = 4$ condition, constant held-out tuples reach $0.912$–$0.931$, whereas mixed tuples reach $0.330$–$0.442$. Because the modular hold-out rule always includes $(0, 0)$, pooled held-out accuracy can therefore overstate recovery of arbitrary unseen combinations.

### D.4 Fusion of Multiple Statistical Evidence Sources

We next test how several statistical observations about the same latent state are combined. Two conditionally independent carriers provide evidence about one shared binary state. Across the experimental battery we vary carrier reliability, whether the sources agree or conflict, context size, and prior probability. This distinguishes a symbolic interpretation (in which decoded carrier states might simply vote) from the posterior account of Equation 3, in which each source contributes according to its learned likelihood.

For two conditionally independent sources $C_1$ and $C_2$, that account gives

$$p(z \mid C_1, C_2) \propto p(z)\, p(C_1 \mid z)\, p(C_2 \mid z). \tag{20}$$

Equivalently, posterior log odds add the prior log odds and the source-specific log-likelihood ratios:

$$\log \frac{p(z = 1 \mid C_1, C_2)}{p(z = 0 \mid C_1, C_2)} = \log \frac{p(z = 1)}{p(z = 0)} + \sum_{i=1}^{2} \log \frac{p(C_i \mid z = 1)}{p(C_i \mid z = 0)}. \tag{21}$$

Frozen models respond causally to both sources and shift their prediction with source reliability. When the carriers agree, their evidence reinforces; when they conflict, the more reliable source exerts greater influence. These observations establish evidence-sensitive fusion behavior, but the stronger test is whether the quantitative weighting is determined by the likelihoods learned during training.

As in Appendix D.2, we therefore compare posteriors computed from the physical test-time rendering with posteriors that score those same rows using each carrier's native training-time likelihood. For the primary mean-plus-covariance battery with a uniform prior, this yields five reliability regimes and five independent model seeds, for 25 models.

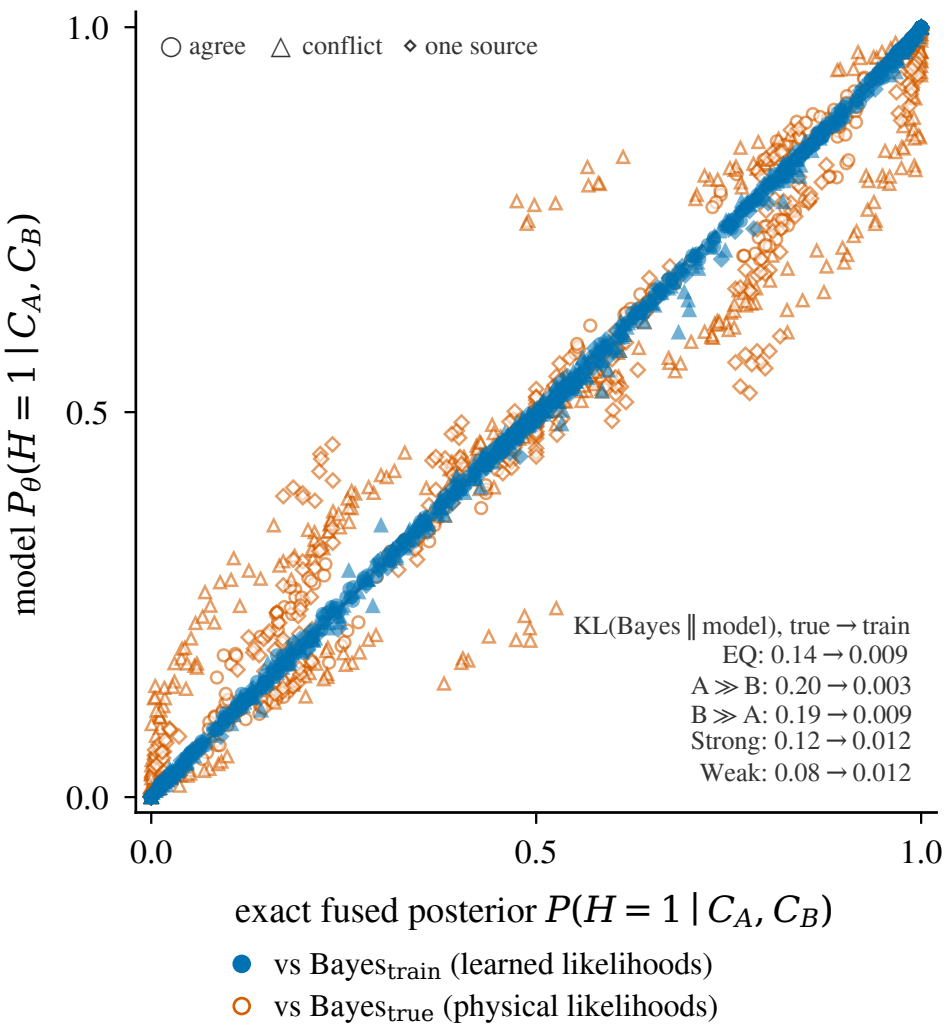



Figure 19: **Multiple statistical observations are fused according to learned likelihood semantics.** Across the two-source mean–covariance battery, predictions are substantially better aligned with the exact fused posterior obtained from the carriers' training-time likelihoods than with the posterior implied by their physical test-time strengths.

Using the physically rendered likelihoods gives pooled KL divergences between $0.06$ and $0.32$ nats across the 25 models (Figure 19). Replacing only those likelihoods by the corresponding training-time likelihoods reduces KL to $0.005$–$0.016$ nats in every model, a $5.5$–$45.7\times$ reduction with median improvement of $18.9\times$. All 25 models then fall below the $0.02$-nat alignment threshold. The model outputs are unchanged between these comparisons; only the external posterior reference is recomputed.

The two carrier families contribute approximately equally per unit of learned evidence. Remaining discrepancies are largest when predictions saturate or when both sources are extremely weak. Accordingly, the experiment supports likelihood-weighted behavioral fusion within this controlled family; it does not establish that the network internally represents or explicitly sums likelihood ratios.

Taken together with the single-carrier intervention of Appendix D.2, the fusion experiment shows that learned evidence semantics persist when several interfaces are present simultaneously. The carrier is therefore not merely decoded into a fixed symbolic state: its quantitative influence depends on the evidential relationship assigned to it by training, and those learned relationships determine how multiple interfaces combine.

### D.5 Training Assigns Authority Between Conflicting Interfaces

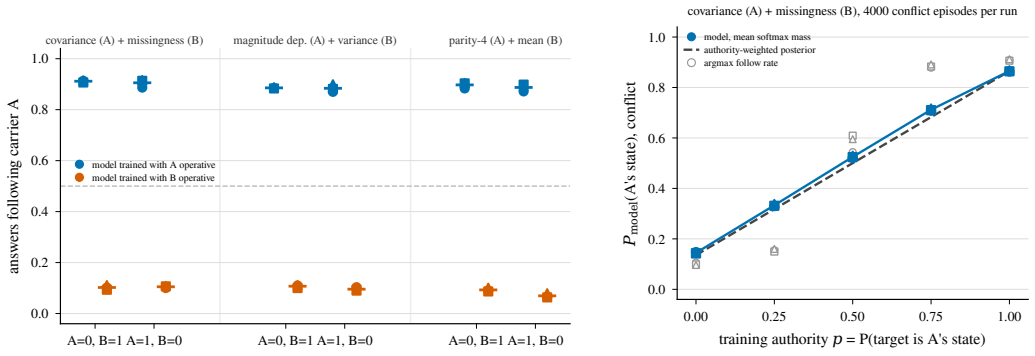
Fusion assumes that two carriers describe the same latent state. We finally ask what happens when two simultaneously present interfaces disagree, and whether training determines which of them controls the prediction.

Both carriers are rendered in every episode at their own calibrated strength and carry independently drawn states, so they disagree in roughly half of all training episodes. For a given pair, two models are trained whose training distributions differ in exactly one respect: which carrier's state is the target. After training, both models of a pair are evaluated on the same forced-conflict episodes. Three carrier pairs, three seeds, and both conflict directions give 36 model-cells, each scored on 600 decisive episodes. Because both members of every pair are individually readable at the same calibrated ideal-observer level (achieved $\mathrm{Acc}^*$ 0.889–0.910, single-carrier read efficiency 0.960–1.005), neither carrier is physically stronger than the other.

Authority follows training. On identical conflicting contexts, each model answers with the state of the carrier its own training made operative in 0.870–0.937 of decisive queries, against 0.063–0.130 for the other carrier; the margin is 0.740–0.873 and the direction is correct in all $36/36$ model-cells (Figure 20a). Reversing which carrier is operative during training therefore reverses the behavior under the same physical conflict. Controls behave as expected: two-carrier models reach 0.971–1.013 of their own ideal observer, agreement-condition accuracy is 0.886–0.961, and permuting context rows changes accuracy by exactly 0.000 in all 18 runs.

Authority is also graded rather than binary. Training one pair with a latent per-episode authority probability $p$ that the target is the first carrier's state, and evaluating 4,000 conflict episodes per run, the model's mass on that carrier's state rises monotonically with $p$ in every seed: 0.143–0.148 at $p = 0$, 0.330–0.338 at $p = 0.25$, 0.521–0.530 at $p = 0.5$, 0.708–0.717 at $p = 0.75$, and 0.863–0.868 at $p = 1$ (Figure 20b). At equal authority the output stays near the intermediate value instead of committing to one carrier.

The pre-specified criterion distinguished an ordered response from a calibrated one, and the frozen verdict is ordered but not calibrated: the response is monotone in all three seeds, but the mean absolute deviation from the authority-weighted posterior reaches 0.063 and so exceeds the 0.05 calibration threshold at 7 of 15 cells, although the difference of means never exceeds 0.033. Two further limits apply. The conflict is in distribution, because the carriers already disagree during training, so this is a result about learned assignment rather than zero-shot conflict resolution. And, as elsewhere, the evidence is behavioral: it does not show how authority is represented internally.




(a) (b)

Figure 20: **Training decides which of two conflicting statistical interfaces controls the prediction. (a)** Under the same forced conflict, models that differ only in which carrier their training made operative follow opposite carriers, in all three tested pairs and both conflict directions. **(b)** With a graded training authority, the mass placed on the first carrier's state increases monotonically with that authority, tracking the authority-weighted posterior in order but not exactly.

# E Extended Results: Knowledge and Structured Computation

## E.1 Statistical Context Can Condition a Task Prior

We test whether a statistical interface can communicate higher-level information about how ordinary observations should be interpreted. The training distribution contains a latent knowledge state $K \in \{0, 1, 2\}$ that selects one of three priors over a separate latent hypothesis $h \in \{0, 1, 2\}$:

$$\begin{aligned} \pi_0 &= (1/3, 1/3, 1/3), \\ \pi_1 &= (0.2, 0.6, 0.2), \\ \pi_2 &= (0.2, 0.2, 0.6). \end{aligned} \tag{22}$$

The state $K$ is rendered only through a statistical carrier block; neither $K$ nor the numerical prior $\pi_K$ is supplied explicitly to the model.

The three hypotheses are noisy threshold functions with

$$\theta = (-0.5, 0.3, 0.7),$$

and

$$p(y = 1 \mid x, h) = \begin{cases} 0.9, & x > \theta_h, \\ 0.1, & x \le \theta_h. \end{cases} \tag{23}$$

Ordinary labelled context observations use $x_i \sim \mathcal{U}(-1, 0.3)$, whereas query inputs use $x^* \sim \mathcal{U}(0.3, 1)$. Consequently, $h = 1$ and $h = 2$ induce identical likelihoods on every ordinary context row but can make different predictions at the query. The statistical carrier therefore does not encode the hypothesis or output directly; it communicates information about which prior over hypotheses is relevant.

For a known state $K$, the exact target is

$$p(y^* \mid x^*, D, K) = \sum_h p(y^* \mid x^*, h)\, p(h \mid D, K), \tag{24}$$

where

$$p(h \mid D, K) \propto p(D \mid h)\pi_K(h). \tag{25}$$

When the statistical carrier is uncertain, the corresponding reference averages these predictions over its posterior,

$$p(y^* \mid x^*, D, C) \approx \sum_K p(K \mid C)\, p(y^* \mid x^*, D, K). \tag{26}$$

**Near-certain statistical knowledge.** We first calibrate the carrier to $\mathrm{Acc}^* = 0.995$. The posterior probability assigned to the rendered state is $0.996$–$0.997$ across seeds. Model predictions closely follow the corresponding $K$-conditioned target, with KL divergence $1.1$–$1.5 \times 10^{-3}$ nats compared with $24.0$–$26.3 \times 10^{-3}$ nats for a reference that ignores the statistical state.

The result is evaluated counterfactually. For each paired intervention, the latent hypothesis, ordinary data $D$, query, and model weights remain unchanged; only the statistical rendering of $K$ is rewritten. The resulting predictive shift reaches $0.95$–$1.00$ of the exact $K$-conditioned shift across the three seeds, comparable to an explicit-state control (Figure 21).

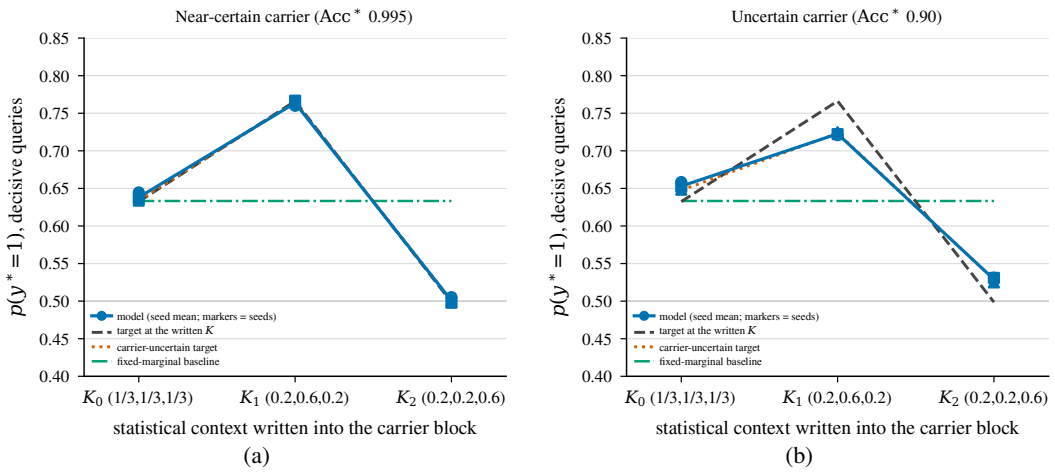



Figure 21: **Statistical context communicates a training-supported knowledge state. (a)** With a near-certain carrier, frozen predictions follow the target conditioned on the written state. **(b)** With an uncertain carrier, predictions are attenuated toward the marginal reference and are better described by averaging over uncertainty in the carrier state.

**Uncertain statistical knowledge.** We next reduce carrier information to $\mathrm{Acc}^* = 0.90$. The posterior probability of the rendered state is then only 0.846–0.860. If the model decoded the most likely state and subsequently treated it as known, the counterfactual shift would remain equal to the hard-$K$ target. Instead, the observed shift falls to 0.71–0.75 of that reference.

When the target is averaged over the uncertainty supported by the carrier as in Equation 26, intervention slopes return to 0.96–1.00 and KL decreases from 7.2–8.9 $\times 10^{-3}$ to 1.0–1.4 $\times 10^{-3}$ nats. This uncertainty-weighted analysis is descriptive rather than a pre-specified verdict criterion (Figure 22).

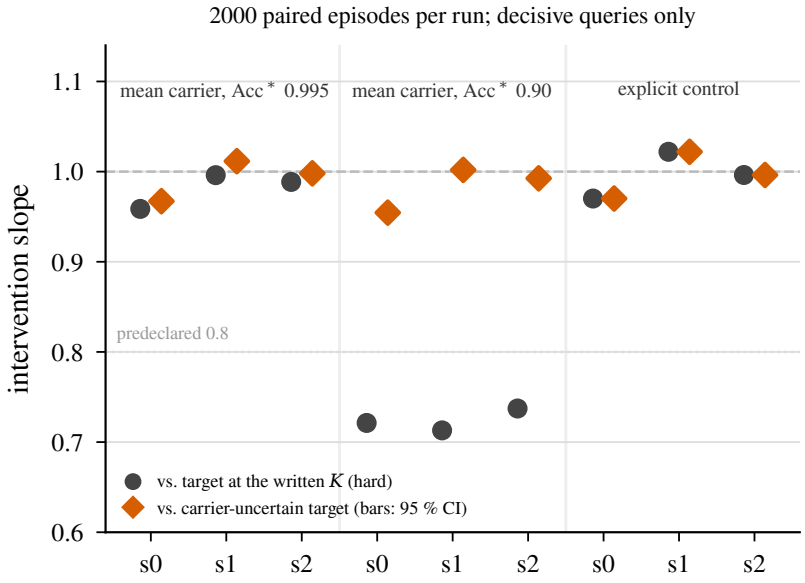



Figure 22: **Counterfactual response tracks the information available about the written knowledge state.** With a near-certain carrier and the explicit-state control, the model closely follows the hard-$K$ intervention target. With an uncertain carrier, the response is attenuated and instead closely matches the target that accounts for carrier uncertainty.

The result should not be interpreted as inference-time modification of the model's learned meta-prior. All three priors are part of the training distribution and model weights remain fixed. The statistical context supplies evidence about which training-supported prior is relevant for the current episode. Nor does this experiment identify an internal hierarchical Bayesian representation; it establishes the corresponding input–output behavior.

### E.2 Statistical Context Can Select Structured Computations

We test structured computation using ordered two-step programs. The scalar query operand is mapped to one of eight equiprobable bins,

$$b = \lfloor 8\Phi(x) \rfloor.$$

Four primitive transformations are fixed permutations of these bins:

$$\begin{aligned} \sigma_0 &= [0, 1, 2, 3, 4, 5, 6, 7], \\ \sigma_1 &= [5, 6, 4, 2, 3, 0, 1, 7], \\ \sigma_2 &= [3, 1, 4, 7, 5, 6, 2, 0], \\ \sigma_3 &= [6, 2, 5, 4, 3, 7, 1, 0]. \end{aligned} \tag{27}$$

A program is an ordered pair $(f_1, f_2)$ and applies

$$b' = \sigma_{f_2}(\sigma_{f_1}(b)),$$

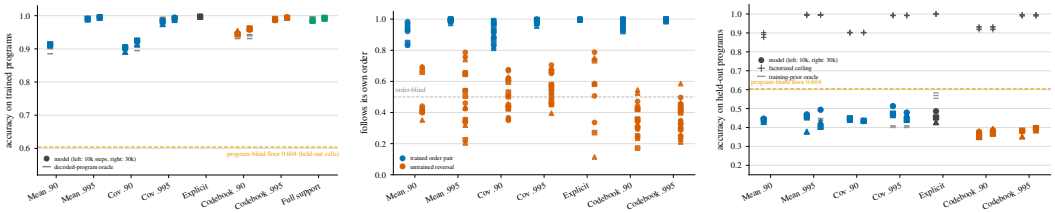
with binary target $y = \mathbf{1}[b' < 4]$.

Each instruction slot is encoded in a separate statistical carrier block. The first block specifies $f_1$, the second $f_2$; the query itself contains only the operand. Thus, exchanging the two carrier states changes the requested order of computation.

**Supported programs are executed.** We first train on 12 of the 16 ordered programs while keeping each primitive equally frequent in each instruction slot. At standard carrier strength ($\mathrm{Acc}^* = 0.90$), supported-program accuracy is $0.911$–$0.914$ for mean carriers and $0.891$–$0.906$ for covariance. With near-unambiguous carriers ($\mathrm{Acc}^* = 0.995$), accuracy reaches $0.975$–$0.991$ at 10k steps and $0.987$–$0.996$ at 30k.

A full-support control trains all $4 \times 4 = 16$ programs. Accuracy reaches $0.984$–$0.986$ at 10k and $0.991$–$0.994$ at 30k, compared with a decoded-program oracle of $0.995$–$0.997$. Thus, every member of the tested program family can be executed when supported during training.

**Instruction order matters.** On query operands for which reversing the two primitives changes the correct answer, an order-blind strategy has follow rate $0.5$. For supported nontrivial orderings, statistical models follow the written order in $0.812$–$1.000$ of decisive cases; with near-unambiguous carriers this rises to $0.956$–$1.000$ at 10k and $0.980$–$1.000$ at 30k (Figure 23).




(a) (b) (c)

Figure 23: **Structured computation through statistical context. (a)** Supported ordered programs approach their decoded-program oracle. **(b)** Models distinguish instruction order on queries where the two orders imply different answers. **(c)** This supported execution does not extend systematically to programs with zero training support.

**Supported execution is not systematic composition.** We then withhold four ordered programs,

$$(1, 2), \qquad (2, 3), \qquad (3, 1), \qquad (0, 0),$$

while retaining the other 12. Every primitive remains represented in both instruction positions, and the reverse orderings of the nontrivial held-out pairs remain supported.

Statistical interfaces nevertheless reach only $0.378$–$0.513$ accuracy on the held-out programs, compared with factorized ceilings of $0.876$–$0.998$. Increasing carrier strength or extending optimization from 10k to 30k does not rescue them.

The failure is not specific to statistical decoding. A factorized explicit interface, which supplies both primitive identities noiselessly, reaches only $0.428$–$0.486$ on the held-out programs. A joint 16-state program codebook reaches $0.349$–$0.397$. Across all statistical, explicit, and codebook conditions and both training budgets, none of 36 valid held-out runs satisfies the pre-specified recombination criterion.

This establishes a distinction between *structured control* and *systematic compositional generalization*. Statistical context can specify an ordered computation that a frozen model learned to execute, but the present experiment provides no evidence that the model learned a factorized interpreter capable of recombining supported primitive instructions into arbitrary unseen programs.

# F Extended Results: Statistical Interfaces in Pretrained Language Models

The language-model experiments use several overlapping model populations: a primary sensitivity screen, a binding screen on a larger set of checkpoints, and three focused follow-ups whose model sets are determined by the outcome of the preceding stage rather than chosen freely. Table 15 lists every checkpoint and the experiments it enters, so that each denominator reported below can be resolved directly.

Table 15: **Frozen language-model checkpoints and the experiments they enter.** ✓ = included; M = included and part of the eight-model matched block-size comparison; N = selected but not evaluable under the tokenizer-matched stimulus construction; – = not run. Parameter scales are the vendor-reported sizes of the released checkpoints.

| | | | Sensitivity and binding | | Realization follow-ups | | |
|---|---|---|---|---|---|---|---|
| Model | Family | Scale | Read | Binding | Block | Decomp. | Rep. |
| qwen3.5:2b | Qwen | 2B | ✓ | ✓ | – | – | – |
| llama3.2:3b | Llama | 3B | ✓ | ✓ | – | – | – |
| gemma3:4b | Gemma | 4B | ✓ | ✓ | – | – | – |
| qwen3.5:4b | Qwen | 4B | ✓ | ✓ | – | – | – |
| qwen3:8b | Qwen | 8B | – | – | ✓ | – | – |
| llama3.1:8b | Llama | 8B | – | – | ✓ | – | – |
| qwen3.5:9b | Qwen | 9B | ✓ | ✓ | M | ✓ | – |
| gemma3:12b | Gemma | 12B | – | – | ✓ | ✓ | – |
| qwen3.5:27b | Qwen | 27B | ✓ | ✓ | M | ✓ | – |
| gemma3:27b | Gemma | 27B | – | ✓ | M | ✓ | – |
| qwen3:32b | Qwen | 32B | – | ✓ | M | ✓ | ✓ |
| qwen3.5:35b | Qwen | 35B | – | ✓ | M | ✓ | – |
| llama3.3:70b | Llama | 70B | – | ✓ | M | ✓ | ✓ |
| qwen3.5:122b | Qwen | 122B | – | ✓ | M | ✓ | – |
| mistral-large:123b | Mistral | 123B | – | ✓ | M | ✓ | N |
| *denominator* | | | 6 | 12 | 11 (8 matched) | 9 | 3 (2 evaluable) |

## F.1 Statistical Sensitivity in Frozen Language Models

We first test whether statistical structure of ordinary demonstration context systematically affects frozen language-model behavior without deliberate interface training. The primary screen evaluates

six models from three model families. No model is fine-tuned, and prompts contain no instruction to identify, estimate, or report a statistical property.

All language-model experiments use a common frozen decoding configuration. Models are served locally through Ollama as raw completions, without a chat template or system prompt. We use temperature 0, a fixed decoder seed of 0, repetition penalty 1.0, a context window of 4096 tokens, and at most 64 generated tokens; top-$p$ and top-$k$ are not overridden from the backend defaults. Context seeds control prompt construction only and do not alter the decoder. All checkpoints use Q4_K_M quantization except `qwen3.5:2b`, which uses Q8_0. Prompt templates, decoding settings, and scoring code are provided in the anonymous repository.

Each prompt consists of ordinary question–answer demonstrations followed by a held-out query. A statistical property of the demonstration set is varied while semantic task content is held fixed as closely as the construction permits. The model freely generates its answer, from which we measure a corresponding behavioral property.

Eight statistical or structural manipulations are tested, spanning marginal frequencies, count and dispersion, trends across demonstrations, and dependencies between answer properties. A lexical manipulation serves as a separate positive control.

A statistical property is counted as read when at least one associated output dimension satisfies all predefined criteria: a 95% confidence interval over context seeds excludes zero in the predicted direction, the effect exceeds a minimum magnitude, it lies beyond the matched null intervention, consistency across seeds exceeds the predefined threshold, and invalid read-outs remain below 30%. The primary cells contain 16 context seeds, two states, and two held-out queries per context.

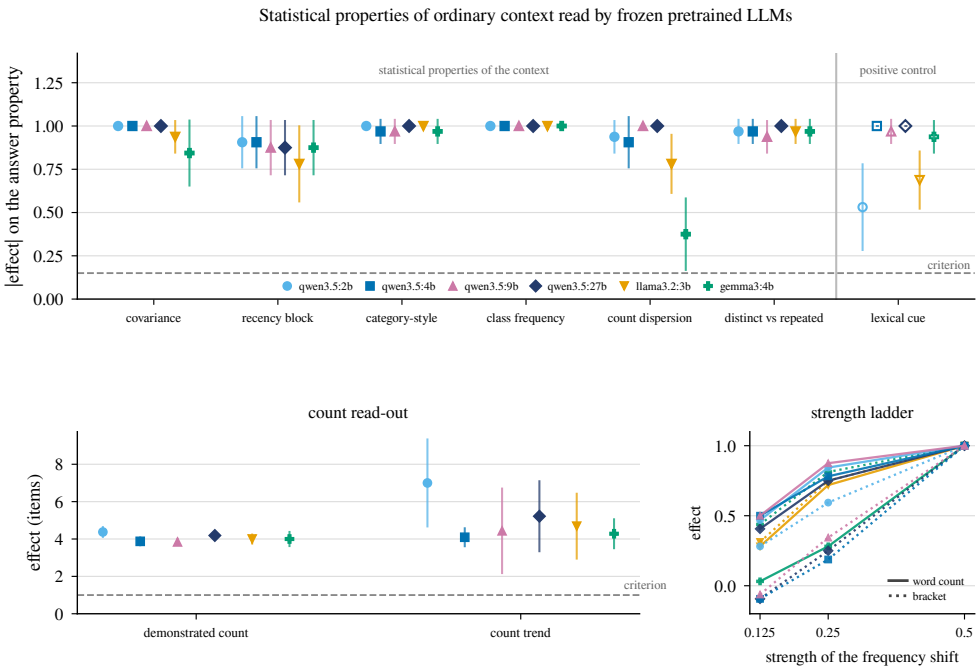



Figure 24: **Frozen pretrained language models respond systematically to statistical properties of demonstration context.** The primary screen contains six statistical manipulations with binary read-outs and two count-based manipulations. The lexical cue is a positive control and is not included in the statistical carrier count.

All eight statistical properties are read in every model tested, yielding $48/48$ successful model–property combinations (Figure 24). A lexical positive control also succeeds in all six models. A fresh-context replication with 32 new context seeds and four queries per context reproduces all $54/54$ tested interfaces including the control.

Not every effect follows the originally hypothesized direction. Across 84 individual read-out cells, 70 follow the predefined direction, 12 show a significant effect in the opposite direction, and two are null; both null cells belong to the lexical control. The most systematic reversal occurs for distinct versus repeated demonstrations: across the relevant channels, models follow the answer style associated with the distinct examples rather than the pre-specified expected direction. We retain these effects as reversals rather than redefining their sign after observing the data.

Several matched controls constrain simpler explanations. Order is separated from recency, dispersion from merely observing an extreme value, and relational structure from its marginal statistics. These controls do not isolate every possible surface property and we therefore interpret this section behaviorally rather than mechanistically.

**Candidate structures that did not yield an interface.** Thirteen candidate context structures were named in the frozen specification before any generation. Nine have a direct behavioral read-out and are the eight statistical properties plus the lexical control reported above. The remaining four did not produce a usable interface and are reported here rather than removed from the battery (Table 16). Two of them vary only the order of demonstration rows, one varies only whether categories appear contiguously while leaving every marginal statistic unchanged, and one requires combining two independently uninformative features. None of the four is systematically read across models, but they fail in different ways: the order-based candidates are inconsistent or null with one opposite-direction cell, contiguity is a clean null, and the higher-order rule is read by three of six models and is therefore inconsistent rather than absent.

Table 16: **Candidate context structures that did not become interfaces.** Counts are models in which the property satisfies all five read criteria on at least one output dimension, out of the six screened models. Largest effect is the biggest absolute per-cell effect observed anywhere in that candidate's cells.

| Candidate | What varies | Read | Largest effect | Outcome |
|---|---|---|---|---|
| Label alternation | order of demonstration rows only | 1/6 | 0.31 | inconsistent |
| Period-4 label pattern | order of demonstration rows only | 0/6 | 0.34 | null; one opposite-direction cell |
| Category grouping | contiguity only; all marginals identical | 0/6 | 0.12 | null |
| Higher-order rule | answer set by the joint value of two individually uninformative features | 3/6 | 0.50 | inconsistent |

**Graded and multi-state sensitivity.** Sensitivity extends beyond two endpoints. A finer class-frequency sweep contains nine statistical states and yields a monotone response in 9 of 10 model–channel series; the remaining series is threshold-like. Independently, varying demonstrated list count over eight states from 2 to 9 changes generated count by approximately one item per step in all five models evaluated.

**Simultaneous readability.** Two statistical properties are next written into the same 16-row demonstration context. Across five models and both same-family and cross-family pairings, intended effects range from $0.34$ to $1.00$. Absolute effects on the other output dimension have median $0.03$ and maximum $0.25$ (Figure 25).

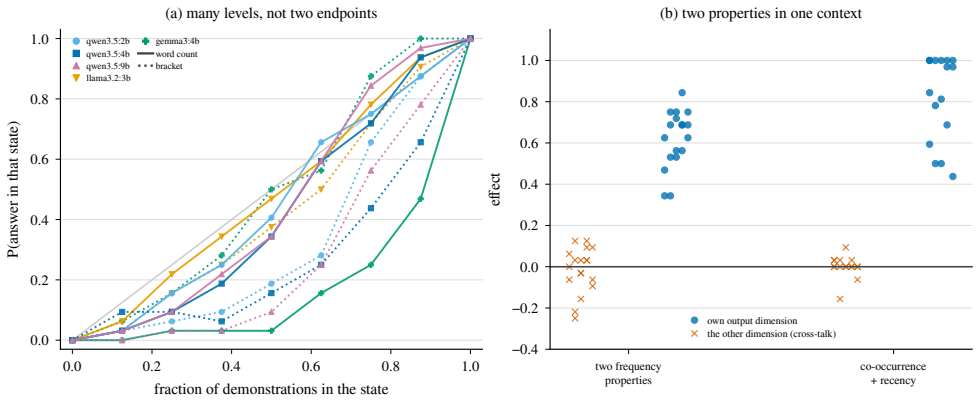



Figure 25: **Resolution and coexistence of statistical sensitivity.** Left: model outputs resolve multiple levels of demonstrated statistical structure. Right: two properties written simultaneously primarily affect their corresponding output dimensions. The coexistence experiment establishes simultaneous readability, not independent latent channels or compositional execution.

These experiments establish *statistical sensitivity*: manipulating statistical structure can systematically change frozen-model behavior. They do not yet establish that the property can acquire an arbitrary newly demonstrated meaning.

### F.2 Statistical Sensitivity Does Not Imply Arbitrary Bindability

We next ask whether in-context demonstrations can assign an arbitrary semantic mapping to a statistical property without changing model parameters.

Each binding context contains demonstrations pairing two physical states of a candidate property with two answer behaviors. The property and its state are never named explicitly. A held-out query contains one of the same two physical states.

The state-to-behavior mapping reverses across context seeds. Thus, the same physical state is associated with opposite answers in different contexts. Successful binding requires the model to infer and follow the mapping demonstrated in the current context rather than relying on a fixed preference.

We test four statistical properties across twelve frozen model checkpoints from four families: block size, repetition versus diversity, order, and relational structure. A lexical nonsense cue replaces the statistical state with one of two arbitrary words and provides a positive control for the in-context mapping task.

A model–carrier cell is counted as binding only when all five fixed criteria are satisfied: the 95% confidence interval excludes zero, absolute effect is at least $0.15$, the effect lies at least three standard errors beyond the matched neutral-block control, per-seed consistency is at least $0.70$, and invalid read-outs do not exceed $30\%$. Invalid generations are counted rather than discarded.

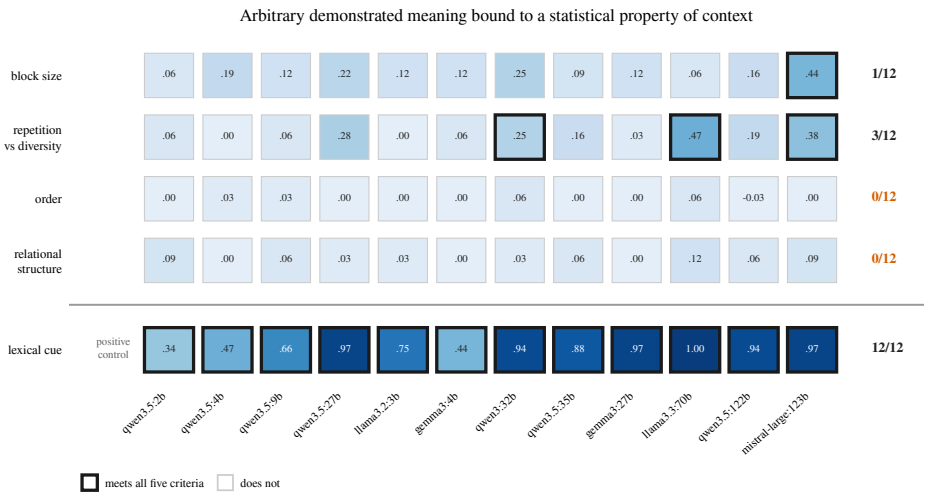



Figure 26: **Arbitrary semantic binding is selective.** Each cell shows one model–carrier combination; outlined cells satisfy all binding criteria. Under the ordinary protocol, block size binds in $1/12$ models, repetition versus diversity in $3/12$, and neither order nor relational structure binds in any model. The lexical positive control binds in $12/12$.

The result separates sensitivity from bindability (Figure 26). Block size, repetition versus diversity, and relational structure are directly readable in all $12/12$ models for which the corresponding direct measurement is available, and order in $11/11$. Yet binding rates under the ordinary protocol are only $1/12$, $3/12$, $0/12$, and $0/12$, respectively.

Relational structure provides a direct dissociation. Its manipulation holds both marginals fixed and changes only their relation. It influences direct behavior in all $12/12$ tested models, but binding effects remain between $0.000$ and $0.125$ and none satisfies the binding criterion. Statistical sensitivity is therefore not sufficient for arbitrary bindability.

Successful cells also respond to both mapping orientations. Because the reported binding effect is oriented by the mapping used in each context, a positive value denotes following the current mapping rather than a fixed association between one carrier state and one answer. The two orientations need not have equal strength, so this test rules out fixed state preference but does not imply behavioral symmetry.

**Strengthening the physical carrier.** Failure in the ordinary protocol does not establish intrinsic unbindability. A subsequent fixed follow-up increases both the number of demonstrations and the contrast of the block-size carrier. At the strongest condition, blocks contain one versus twelve items across 16 demonstration episodes.

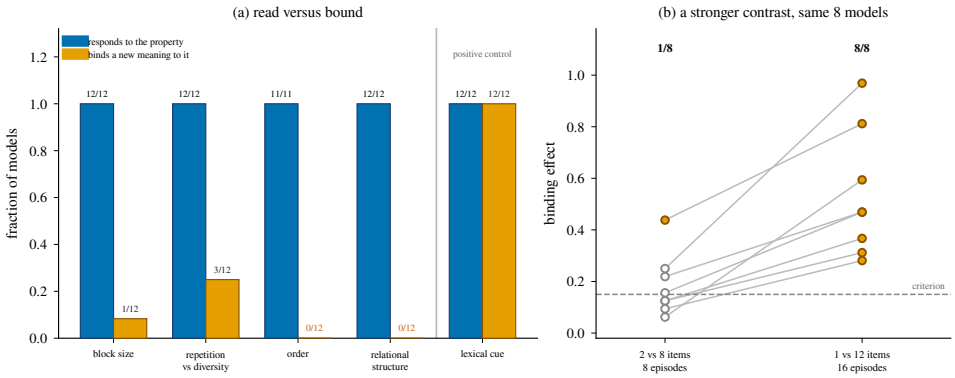



Figure 27: **Readability and arbitrary bindability are distinct.** Left: the candidate properties are broadly readable but selectively bindable. Right: among the eight models evaluated under both block-size contrasts, binding rises from $1/8$ under the ordinary condition to $8/8$ under the stronger realization.

Among the eight models evaluated under both conditions, block-size binding rises from $1/8$ under the ordinary contrast to $8/8$ under the stronger contrast (Figure 27). Across the complete strong-condition model set, binding occurs in $10/11$ models. Because the two screens contain different model sets, the matched $1/8 \rightarrow 8/8$ comparison is the cleaner evidence.

This follow-up was specified after the ordinary binding screen, although its conditions and outcome rule were fixed before its own model inference. It therefore establishes that failure under the original realization is not an impossibility result, but does not by itself identify the physical property responsible for successful binding.

### F.3 Bindability Depends on the Physical Realization

The stronger block-size manipulation changes several correlated physical properties, particularly item count and block extent. We therefore decompose these factors while keeping the binding task, mapping reversal, scoring procedure, and decision criteria fixed.

Across nine frozen models, the reference contrast changes item count from one to twelve and increases block length by an average of $24.7$ tokens. Two alternative constructions approximately match block length while changing item count. A third fixes item count at three while increasing block length by an average of $21.1$ tokens.

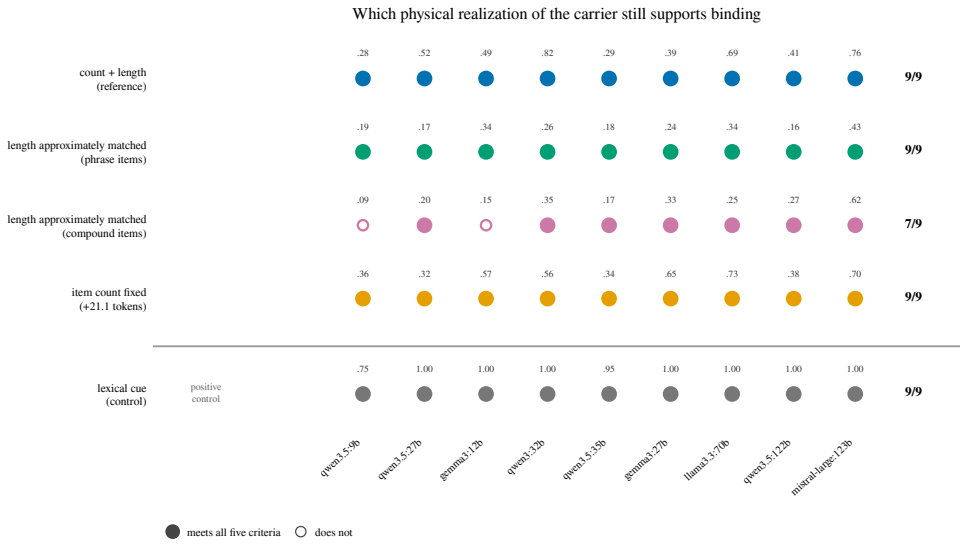



Figure 28: **Binding persists under several physical realizations of the block-size carrier.** Two approximately length-matched count manipulations bind in $9/9$ and $7/9$ models. A count-matched construction with a substantial block-length difference binds in $9/9$. These contrasts rule out simple necessity claims but do not identify one unique causal feature.

Both approximately length-matched constructions remain bindable, in $9/9$ and $7/9$ models (Figure 28). Conversely, fixing item count while changing block length by approximately 21 tokens yields binding in all $9/9$ models. A substantial change in block extent is therefore sufficient for binding in this construction, but block length is not necessary; similarly, item count can support binding when length is approximately matched but is not necessary either.

The constructions retain residual physical differences, including punctuation, lexical form, or adjective use. They therefore do not establish that item count or length is itself the causal variable.

**A tightly matched repetition–diversity follow-up.** We additionally construct a repetition-versus-diversity contrast that matches item count, punctuation, whitespace, casing, character count, byte count, and token count exactly between states. Three models are selected by a frozen rule; one fails a tokenizer feasibility gate before generation and is not counted as a negative result, leaving two technically evaluable models.

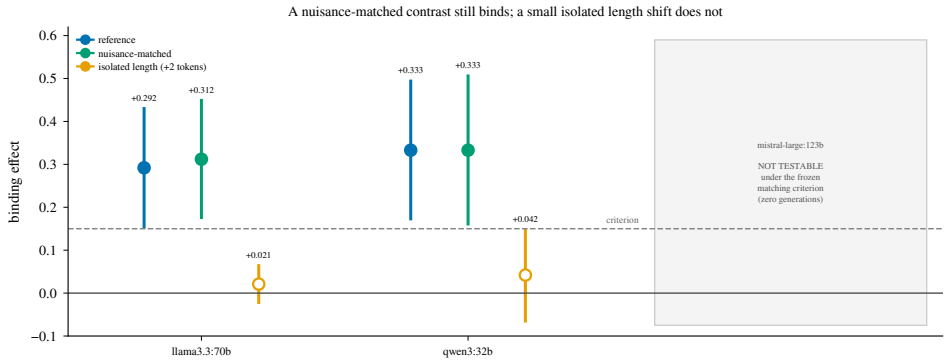



Figure 29: **Binding survives tighter nuisance matching.** For both technically evaluable models, the nuisance-matched repetition-versus-diversity carrier retains arbitrary binding. A separate two-token length manipulation with repetition and item count fixed does not bind, showing that physical magnitude also matters.

The nuisance-matched repetition-versus-diversity carrier retains binding in both evaluable models (Figure 29). For `llama3.3:70b`, the matched effect is $0.312$ (95% CI $[0.176, 0.449]$); for `qwen3:32b`, it is $0.333$ ($[0.161, 0.506]$).

A separate small length-only manipulation changes block length by exactly two tokens while holding repetition and item count fixed. It does not satisfy the binding criterion in either model, with effects $0.021$ and $0.042$. This does not contradict the count-matched 21-token result; together they show that both physical realization and magnitude matter.

Across these experiments, no single tested scalar property explains all successful cases. Block length is not necessary, item count is not necessary, and binding can survive exact matching of several surface quantities. The experiments establish that statistical bindability is selective and realization dependent.

Finally, the LLM experiments stop at behavioral binding. They do not establish that a newly bound statistical state selects and executes a distinct computation in the stronger EXECUTE sense studied in the controlled PFNs, nor do they establish that pretrained LLMs implement the posterior mechanism characterized in Section 3.