WHAT DOES LANGUAGE-MODEL PRETRAINING CONTRIBUTE TO TIME-SERIES UNDERSTANDING?

Anonymous authors
Paper under double-blind review

ABSTRACT

Our contribution is a controlled empirical attribution analysis of the role of language-model pretraining in time-series understanding across decision targets and output interfaces. Across ten base language models, we compare frozen pretrained backbones with architecture-matched random controls under matched inputs, feature extraction, and learned-readout training. In controlled forecasting, pretraining improves process-family routing in all ten models, with gains of 7.5-31.4 percentage points, but the effect reverses when supervision targets the future-optimal expert; a five-model expected-risk control preserves this reversal. Affine forecasting heads instead reduce error in nine of ten models. On four real-data domains, task-specific readouts improve correct updating after evidence edits by 14.80 points on average, while preservation of unchanged facts decreases by 7.59 points. Under native constrained decoding, models retain the pre-edit answer in 88 of 100 evaluations conditioned on an initially correct target. Readout controls show that the measured contribution depends on how temporal features are accessed. Together, the analysis shows that pretraining supplies reusable temporal features, but its benefit is decision- and interface-dependent rather than a uniform gain in time-series understanding.

1 INTRODUCTION

Language-model pretraining is increasingly used as a general substrate for time-series systems. Yet forecasting a signal, identifying its structure, and answering questions about observed evidence are distinct capabilities: recognizing periodicity need not identify the lowest-error predictor, and an answer that is correct once need not update after the relevant evidence changes.

This paper asks an attribution question: which time-series decisions are improved by pretrained weights when architecture, inputs, and readout training are held fixed? A time-series system combines a backbone, a numerical representation, supervision, and an output interface. Learned heads and native decoders use temporal information differently (Gruver et al., 2023; Jin et al., 2024; Xie et al., 2024; Kong et al., 2025). WaveTLM further separates task specification from execution by compiling requests and visible evidence into typed task states (Chen et al., 2026). Because interface choices affect performance, complete-system comparisons cannot by themselves determine what pretraining contributes.

The paper's primary contribution is a controlled empirical attribution analysis of pretrained temporal representations that uses architecture-matched random controls to isolate the contribution of pretrained weights, then varies the decision target and output interface under common inputs, feature extraction, and head training. Figure 1 organizes this analysis around structural recognition, numerical prediction, and question answering, testing whether the same representation supports distinct decisions.

The analysis rejects a uniform view of pretraining benefit. On the same unseen compositions, process-family routing gains range from 7.5 to 31.4 percentage points across ten models, yet future-optimal expert supervision reverses the effect in every model. A five-model expected-risk control preserves the reversal, while affine numerical heads reduce forecasting error in nine of ten models.

Real-data questions reveal a second separation. Across ten models and four domains, task-specific readouts improve correct updating by 14.80 percentage points, whereas preservation falls by 7.59 points. Across five native models, 88 of 100 model-family evaluations with an initially correct target

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.

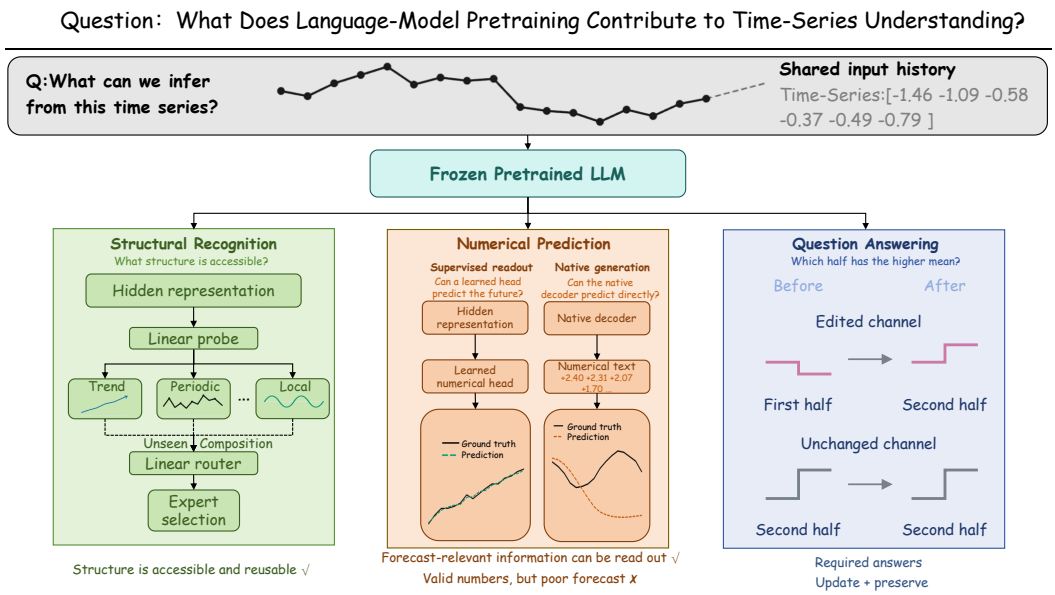

Figure 1: **Three complementary views of time-series understanding. Left:** linear probes test whether temporal structure is accessible in frozen representations, while routers test whether it can support expert selection on unseen compositions. **Middle:** numerical prediction separates supervised forecasting readouts from native generation of numerical text. **Right:** paired edits test whether answers update when the queried fact changes and remain correct when that fact is preserved. In the illustrated half-mean comparison, the target answer changes from first to second half, whereas the control answer remains second half. These are required answers, not model predictions. Structural and numerical tasks use controlled histories; question answering uses real observations and edited copies. Learned readouts compare matched pretrained and random backbones, while native experiments characterize pretrained question answering. The shared-input waveform and forecast curves are schematic.

retain the pre-edit answer. Each question is evaluated independently, so this pattern reflects responses to two evidence states rather than answer history.

Together, the analysis distinguishes structural accessibility, predictive utility, and evidence responsiveness. A common forecasting test isolates the supervision target, while paired real-data edits test whether answers change when the queried fact changes and remain stable when it does not. These comparisons directly characterize how pretrained representations contribute to structure recognition, future prediction, and questions about observed evidence.

2 Related Work

Language models support forecasting through numerical text continuation or task adaptation (Gruver et al., 2023; Zhou et al., 2023; Jin et al., 2024; Liu et al., 2024). The ablations of Tan et al. (2024) ask how much forecasting performance depends on pretrained weights. We turn this attribution question into a matched comparison: pretrained and random backbones share the architecture, histories, and readout training, while the supervised routing target changes under a common future-defined evaluation.

Feature-understanding and multi-task question benchmarks test diverse properties of observed signals (Fons et al., 2024; Kong et al., 2025). ChatTS aligns temporal signals with language using synthetic data (Xie et al., 2024), while WaveTLM compiles requests and evidence into typed task states for numerical and symbolic execution (Chen et al., 2026). We connect these lines by isolating pretrained weights across changes in supervision, readout, and evidence. Building on behavioral testing and contrast sets (Ribeiro et al., 2020; Gardner et al., 2020), paired edits jointly test correct updating of changed facts and preservation of unchanged facts.

The readout is part of the measured system. Probe-control studies show that a learned classifier must be interpreted together with its supervision and capacity (Hewitt & Liang, 2019; Belinkov, 2021). Recent time-series work likewise contrasts prompting with representation-based classification and studies the geometry of transfer (Schumacher et al., 2026; Roger et al., 2026). We therefore report task-specific readouts, a question-conditioned scorer, and native answers separately. Appendix C compares Chronos, TimesFM, MOMENT, and Moirai, whose time-series pretraining uses different architectures and numerical interfaces (Ansari et al., 2024; Das et al., 2023; Goswami et al., 2024; Woo et al., 2024). Appendix H develops the broader methodological context.

3 Experimental Framework

3.1 What the Comparisons Hold Fixed

Controlled forecasting evaluates routing and numerical prediction with ten language models, with a five-model expected-risk control. Real-data question answering uses ten models for learned readouts and five for native answers. All are base checkpoints. Appendix A gives the complete coverage table, model list, extraction rules, and optimization settings.

Learned-readout comparisons fix the architecture, tokenizer, histories, splits, feature extraction, and head training; only the frozen backbone changes between released pretrained weights and architecture-native random initialization. Figures and tables use **P** for pretrained and **R** for random, so P-R isolates the pretraining contribution within each architecture. The random backbone receives the same supervised head training and serves as a learned architectural control.

Positive P-R accuracy differences favor pretraining and are expressed in percentage points. Prediction results report absolute mean squared error and relative reduction. Seeds 7, 17, and 27 index three repeated fits on the fixed data; they measure fitting variation rather than independent variation in the dataset or pretraining process.

3.2 Structure and Future Values

Each synthetic example has 64 observed values and a 16-value future. The processes include trends, periodic signals, local autoregressive signals, mixtures, and regime changes. Values are normalized using the context alone and serialized as signed decimals with two places. We extract the last-layer state of the final nonpadding token. Appendix B specifies the generators and preprocessing.

A linear router selects among three fixed predictors: trend extrapolation, seasonal prediction, and fifth-order autoregression. It trains on 270 windows from primitive processes and is evaluated on 120 unseen-composition windows, balanced by the expert that best predicts the future. These compositions are absent from training, making this an out-of-distribution test.

The routing comparison separates process recognition from future-optimal decision making. Process-family supervision maps a trend, periodic, or local process to its corresponding expert. Future-optimal expert supervision instead selects the expert with the smallest mean squared error on that example's realized future:

$$j^\star(x, y) = \arg\min_j \mathrm{MSE}(f_j(x), y). \tag{1}$$

Here MSE denotes mean squared error, f_j an expert's forecast, x the history, and y the realized future. Both routers are evaluated against the same future-defined winner; the comparison changes only the supervision that shapes the router.

To distinguish the supervision effect from dependence on one noisy future, an expected-risk control holds the latent process and context fixed, samples 20 future-noise realizations, and labels the context by the expert with the lowest mean forecast error across them:

$$j^{\mathrm{risk}}(x) = \arg\min_j \frac{1}{20} \sum_{k=1}^{20} \mathrm{MSE}(f_j(x), y^{(k)}). \tag{2}$$

The label is defined by mean loss rather than the most frequent winning expert, and the five-model control uses its own matched reference fits. The numerical branch then asks a different question: an affine head predicts all 16 future values directly from frozen features. With 450 training and 300 test windows per seed, it measures the forecast-relevant information accessible to this readout.

3.3 Questions About Real Observations

The question dataset contains 264 source groups, with 66 from each of four domains: PTB-XL electrocardiograms, Melbourne pedestrian counts, hourly electricity demand, and Australian temperature and rain observations. PTB-XL groups respect patient identity and the dataset's official folds (Wagner et al., 2020). Fixed source-group hashing assigns splits in the other domains. The resulting train, validation, and test sets contain 193, 29, and 42 groups. All questions and edits from a group stay in its split.

Each group provides one observation window and six questions about its trend, half means, peak location, control statistic, joint description, and conditional peak location. Repeating target and control questions after an edit yields eight items per group, 2,112 overall and 336 in test. Models receive timestamps, channel names, units, and values rounded to four decimals. Gold answers use these visible values. Appendix D gives the construction details.

The edit defines a paired test of evidence use. Suppose the first half of a signal has the higher mean. Adding a sufficiently large constant to every value in the second half reverses that relation, so the target answer must change. A control question concerns either an untouched channel or a within-half range preserved by a constant offset, so its answer must remain correct. The four required answers are

	Target question	Control question
Original window x	a	b
Edited window $T(x)$	$a' \neq a$	b

(3)

We call this set an *evidence family*. Every test family satisfies target change and control preservation after numerical serialization. Questions and option order remain identical within each original-edited pair, and every question is evaluated independently. The edited observations are numerical interventions that test responses to the supplied evidence, rather than claims about naturally occurring physiological or environmental events. Appendix G summarizes validation of question wording, visible-value golds, and evidence-family predicates.

3.4 Three Ways to Answer a Question

Task-specific readout. A separate linear classifier is trained for each semantic task on the frozen temporal representation. It receives an explicit task identity and no question text. Pretrained and random conditions share head initialization and a 100-epoch training budget.

Question-conditioned scorer. A shared multilayer perceptron scores each candidate using temporal features and an encoding of the question and candidate. Both inputs are projected to 128 dimensions; the hidden layer has width 64. The question encoder is pretrained in both conditions, so P-R changes only the temporal branch. Both temporal conditions use the same head initialization, minibatch order, and 100-epoch training budget.

Native answer. Five pretrained models answer directly through two protocols. Free generation uses deterministic decoding and a strict option-letter parser; invalid responses count as incorrect. Constrained choice selects the allowed answer with the highest continuation likelihood, removing invalid formats by construction. Neither protocol trains a task head. These results characterize the pretrained decoder without comparing it with random weights.

Readout and edit controls use the same source groups and are described with their results in Section 4.5 and Appendix F.

3.5 Scoring Correct Updating and Preservation

A useful answer change must be correct. We count a successful update only when both the original and edited target answers are correct; preservation likewise requires both control answers to be correct. Joint success requires all four answers in an evidence family. Writing $C_{t0}, C_{t1}, C_{c0}, C_{c1}$ for their binary correctness indicators gives

$$\begin{aligned} \mathrm{TUS} &= \langle C_{t0} C_{t1} \rangle, & \mathrm{CIS} &= \langle C_{c0} C_{c1} \rangle, \\ \mathrm{FCJS} &= \langle C_{t0} C_{c0} C_{t1} C_{c1} \rangle. \end{aligned} \tag{4}$$

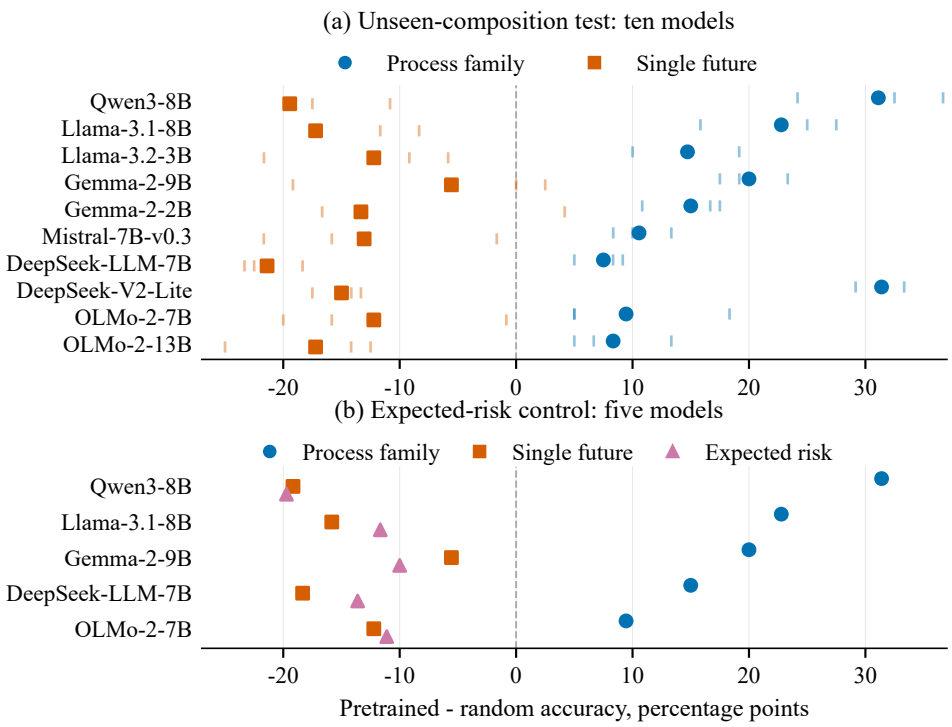

Figure 2: **Supervision changes the sign of the pretraining effect.** Panel a shows ten-model differences in balanced routing accuracy on the common unseen-composition test. Panel b shows a separate five-model control using 20 future draws. All three targets in panel b use that control's own reference fits. Values are pretrained accuracy minus random-backbone accuracy in percentage points; positive values favor pretraining. Faint ticks in panel a show the three seed effects.

TUS denotes target update success, CIS control invariance success, and FCJS four-cell joint success. The figures use *updating*, *preservation*, and *joint success*. These paired metrics distinguish systems with equal item accuracy: correct items may form complete successful families or be distributed across families that each miss a required answer.

We average paired scores over source groups within each domain and then give the four domains equal weight. Item accuracy also weights tasks equally, so repeated target and control questions do not double their contribution. Learned-readout results average three seeds within each model and, where stated, the fixed model set. Every model uses the same 42 test groups. Seed marks show variation across fits, rather than confidence intervals. Appendix E documents these definitions and reports the paired outcomes separately for each domain.

4 Results

4.1 Routing Effects Reverse with the Supervision Target

Pretraining benefits expert routing when supervision aligns with process identity. Under process-family supervision, all ten models improve on the unseen-composition test, with gains of 7.5-31.4 percentage points in Figure 2a. When the same histories are instead labeled by the expert with the lowest error on the realized future, the pretrained-random effect reverses in every model, ranging from -21.4 to -5.6 points. The histories, expert bank, and future-defined evaluation remain fixed; only the training target changes.

The reversal persists when labels average over future noise. Averaging loss over 20 future draws reduces label variability, yet the expected-risk effects remain negative, from -19.7 to -10.0 points, for all five models in Figure 2b. Pretraining therefore makes process structure more accessible without providing the same advantage for selecting the predictor with the smallest future error.

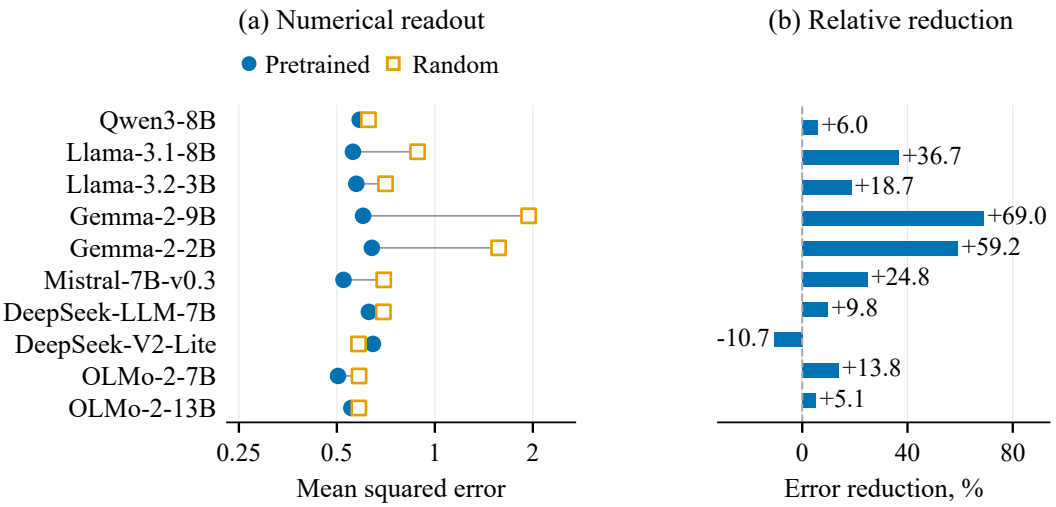

Figure 3: **Numerical readouts improve in nine of ten models.** Panel a gives mean squared error for pretrained and random backbones, averaged over three seeds and shown on a logarithmic axis. Panel b gives the relative reduction $100(\overline{\mathrm{MSE}}_R - \overline{\mathrm{MSE}}_P)/\overline{\mathrm{MSE}}_R$. Positive reductions favor pretraining. The model order matches Figure 2a.

Appendix B reports the corresponding absolute accuracies. The matched effects isolate how pretraining changes each routing objective rather than ranking the two pretrained routers against one another. Appendix C further shows that the family-supervised mean gain remains positive at every tested projection width, so the structural effect is not tied to the original feature dimension. We next test whether the same features support numerical forecasting directly.

4.2 Numerical Readout Shows a Distinct Pretraining Effect

Pretrained features support supervised numerical prediction in nine of ten models. An affine head has lower mean squared error than its random-backbone counterpart in these models. Figure 3 reports both absolute errors and relative reductions; Gemma-2-9B shows the largest reduction, from 1.94137 to 0.60180, or 69.0%. Numerical gains therefore remain widespread even when routing effects depend on the supervision target.

The separation is especially clear for DeepSeek-V2-Lite: it has the largest family-routing gain, yet its affine numerical error rises from 0.58273 to 0.64530, a 10.7% increase. Model order under structural routing therefore does not predict model order under numerical readout. A nonlinear numerical head improves this model in Appendix C, further showing that predictive utility depends jointly on the representation and the readout used to access it.

Routing and numerical prediction characterize information recovered by supervised heads. The real-data experiment next asks whether answers select the relevant information on both sides of an evidence edit.

4.3 Task-Specific Readouts Improve Updating, Not Preservation

Pretraining changes how task-specific readouts respond to edited evidence. All ten models have positive mean pretrained-random differences in correct updating, which rises from 10.94% to 25.74% on average. Preservation moves in the opposite direction, from 47.30% to 39.70%, with negative effects in nine of ten models. Domain-level effects vary: biomedical updating moves from 15.71% to 10.48%. Figure 4 places these effects beside joint success, which rises from 6.47% to 8.40%.

The paired effects are largely invisible to ordinary item accuracy. Across all eight items, task-specific readouts score 48.29% with pretraining and 48.78% with random backbones, close to the 48.31% task-semantic majority reference; uniform option choice has an expected accuracy of 41.67%. Yet

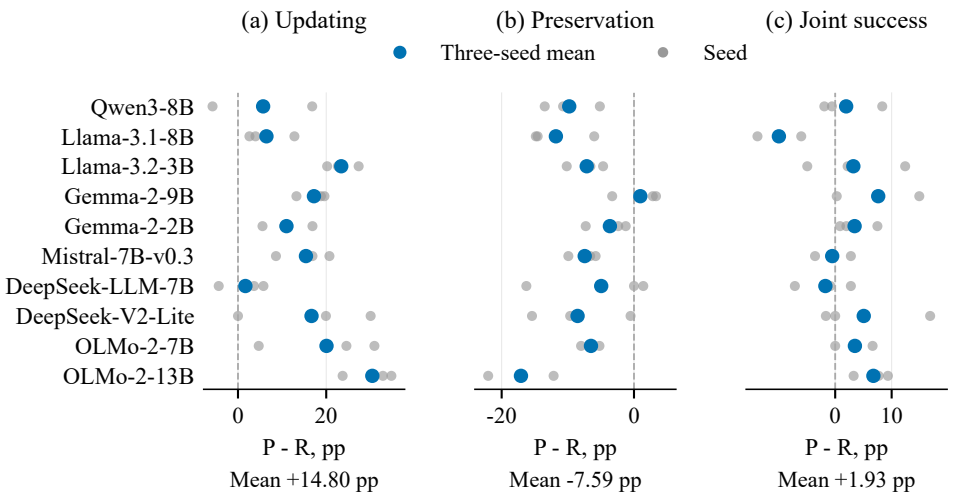

Figure 4: **Better updating does not imply better preservation.** Task-specific readouts use the same 42 test source groups in all panels. Blue points show mean P-R differences over three seeds; gray points show seed differences. The axis unit pp denotes percentage points. Scores weight domains equally. Table 1 gives absolute performance.

Table 1: **Absolute question-answering results.** All scores are percentages. P denotes pretrained weights and R random initialization. Learned readouts average three seeds within each model, then ten models; native protocols use five models. Acc. is all-item accuracy, Update and Preserve require both answers in the corresponding pair, Joint requires all four answers, and Valid is the legal-output rate. Each domain receives equal weight.

Answering interface	Weights	Models	Acc.	Update	Preserve	Joint	Valid
Task-specific	P	10	48.29	25.74	39.70	8.40	100.00
	R		48.78	10.94	47.30	6.47	100.00
Question-conditioned	P	10	47.32	0.22	46.64	0.16	100.00
	R		47.88	1.37	48.67	0.37	100.00
Native free	P	5	16.78	2.48	22.18	1.71	41.24
Native constrained	P	5	41.61	5.27	42.74	3.02	100.00

semantic answer changes rise from 16.03% to 38.65% for targets and from 1.88% to 11.86% for controls. Appendix E.4 separates correct and incorrect transitions and reports the complementary subset in which both questions compare half means.

The same evidence families produce a different result under the shared question-conditioned scorer: mean updating is 0.22% with pretrained temporal features and 1.37% with random features. The task-specific gain is therefore not reproduced by the shared question-conditioned interface. Table 1 places both readouts on the same scales, and Appendix E reports every model and seed. We return to this interface dependence after examining native answers.

4.4 Native Decoders Often Retain Pre-Edit Answers

Native constrained choice reveals substantial pre-edit answer retention without requiring a parseable generated string. Across the five models, the original target answer is correct in 100 model-family evaluations; after editing, 88 retain the semantic answer associated with the original evidence and 12 select the new correct answer. Figure 5a reports each model's counts. These counts condition on an initially correct target and include repeated evaluations of the same families by different models. Appendix E.4 separates edit direction and original correct option position.

Joint success imposes the strictest requirement because it combines correct updating with preservation of the unchanged fact. The counts are 1, 3, 0, 1, and 2 of 42 families for Qwen3-8B, Llama-3.1-8B, Gemma-2-9B, DeepSeek-V2-Lite, and OLMo-2-7B, respectively. Each question is evaluated

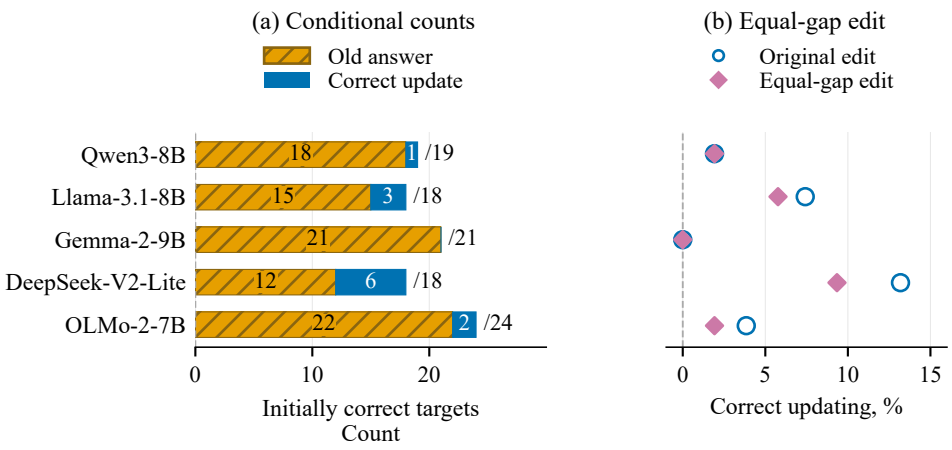

Figure 5: **Native answers often retain the answer to the original evidence.** Panel a counts constrained-choice outcomes only where the original target answer was correct; each bar shows its own denominator. Panel b gives correct updating over all 42 test families, with equal domain weights, for the original and equal-gap edits. Panel a reports conditional counts, whereas panel b reports percentages over the full set of test families.

independently, so pre-edit answer retention describes the relation between predictions on two inputs rather than conversational history or an internal-memory claim.

Free generation further separates output validity from evidence responsiveness. On the 168 core items, Gemma and DeepSeek produce no legal responses under the strict parser, whereas Qwen and OLMo produce legal responses for every item. Appendix E reports these results alongside constrained choice. Producing a legal answer and selecting the answer supported by edited evidence are distinct requirements.

The equal-gap control tests whether changing the half-mean gap drives the result. It reverses the relation while preserving the original absolute gap. On the same families, Qwen and Gemma's updating scores remain unchanged, while the other three point estimates decrease (Figure 5b). Thus matching the gap does not increase updating, although the visible values necessarily change.

4.5 Pretraining Gains Depend on the Readout

The visible-statistics positive control establishes that the paired questions are learnable under the shared scoring architecture. Replacing temporal features with 20 visible statistics yields updating scores of 34.20%, 47.08%, and 44.21% after 100 epochs for the Qwen, Llama, and Gemma question encoders; at 500 epochs, they reach 47.24%, 66.54%, and 60.14%. Figure 6c compares the separately trained budgets. The near-zero updating of the original shared scorer therefore cannot be attributed to an unlearnable question format.

The benefit recovered from model features also changes with the readout. Projecting the task-specific input to 128 dimensions lowers updating from 15.79% to 0.64% for Qwen, from 34.77% to 14.08% for Llama, and from 21.16% to 12.59% for Gemma. Giving the question-conditioned scorer separate task heads raises Llama from 0.93% to 6.70% and Gemma from 0.64% to 6.89%, while Qwen remains at zero. The readout is therefore part of the attribution claim, not a neutral wrapper around the representation.

Together, the controls show that the measured contribution depends on the interaction between representation and readout. Projection also changes normalization, while separate heads add task identity and parameters; these variants therefore establish interface dependence without isolating one factor. Question-only updating is zero by construction because identical questions and options have opposite target answers before and after the edit. Appendix F reports matched controls, learning curves, and a capacity diagnostic.

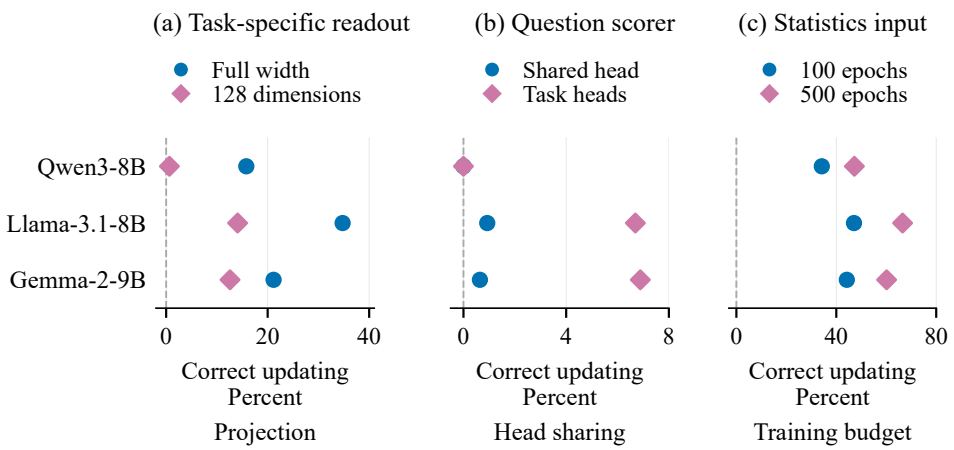

Figure 6: **Updating depends on the readout and its input.** Panel a compares full-width and projected task-specific readouts. Panel b compares shared and task-specific question-conditioned heads. Both use pretrained temporal features. Panel c replaces those features with 20 visible statistics and compares two training budgets; model names identify the question encoder. All points average three seeds. The budget points are separate fitted conditions, not samples from one training trajectory.

5 Discussion

The results reject a scalar view of pretraining benefit. The same pretrained backbone can make process structure more accessible, lose that advantage under future-optimal expert selection, and still support lower numerical forecasting error. Evidence-sensitive behavior likewise appears under a task-specific readout but not under a shared question-conditioned scorer. Pretraining is therefore better understood as a decision-conditioned resource than as a uniform increase in time-series competence.

This distinction changes how time-series representations should be evaluated. Structural accessibility, predictive utility, and evidence responsiveness are not interchangeable. A probe measures what its supervision and readout can recover, but not whether the same representation supports another decision target or output interface. Paired evidence families add a complementary requirement: answers must change when the queried fact changes and remain stable when it does not.

Each interface should therefore be evaluated on its intended use: numerical prediction, expert selection, or responses to controlled evidence changes. Within the stated models, inputs, expert bank, and training budgets, the matched controls identify when pretrained weights alter behavior under fixed architecture and data conditions. The real-data analysis extends this attribution to 42 source groups across four domains through a half-window offset edit, characterizing behavioral contributions without making a claim about a shared internal mechanism.

6 Conclusion

Language-model pretraining provides useful temporal representations, but its benefit is neither uniform nor interface-independent. Across matched pretrained and random backbones, process-family routing improves in all ten models, whereas future-optimal expert supervision reverses the effect; affine numerical readouts nevertheless reduce forecasting error in nine models. On real observations, task-specific heads improve correct updating while preservation declines, and native decoders frequently retain the pre-edit answer. These results distinguish structural accessibility, predictive utility, and evidence responsiveness as distinct properties of a time-series system. More broadly, our analysis shows that the contribution of pretraining cannot be characterized by a single downstream score, because different decisions expose different aspects of the same frozen representation. Evaluating pretrained backbones therefore requires more than aggregate forecasting or question-answering accuracy: the supervision target, readout, output validity, and behavior under controlled evidence changes are all part of the empirical measurement.

AI USE STATEMENT

Generative AI assisted with manuscript organization and drafting, literature lookup, interpretation of experimental results, implementation of score-recalculation and plotting scripts, and LaTeX editing. Numerical figures and tables were generated by executable code from saved experimental outputs, without additional model training or inference. Responsibility for the final scientific content remains with the authors.

ETHICS STATEMENT

This study analyzes synthetic sequences and public time-series observations. PTB-XL contains de-identified electrocardiograms; the questions concern visible numerical relations, not diagnoses or clinical decisions. Its illustrated excerpt is attributed under CC BY 4.0. Derived-real edits are explicitly identified and are not presented as recorded or physiologically validated signals. Source groups remain disjoint across splits. Appendix D documents each source, its attribution requirements, and the scope of the redistributed material.

REPRODUCIBILITY STATEMENT

The accompanying artifact contains the source of this manuscript, six vector figures, their plotting code and numerical inputs, per-item semantic predictions, source-group reconstruction metadata, verification reports, and build instructions. Appendix G specifies the finalized data construction, aggregation rules, and validation checks. All reported scores, descriptive analyses, and figures are reproduced from saved outputs without additional training or model inference. Rebuilding the manuscript and figures requires no checkpoint downloads. Reconstructing the underlying observations requires the original public sources and their applicable access terms.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48:207–219, 2021. URL https://api.semanticscholar.org/CorpusID:236924832.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society series b-methodological*, 57: 289–300, 1995. URL https://api.semanticscholar.org/CorpusID:45174121.

DeepSeek-AI Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, et al. Deepseek llm: Scaling open-source language models with longtermism. *ArXiv*, abs/2401.02954, 2024. URL https://api.semanticscholar.org/CorpusID:266818336.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza, Max Mergenthaler-Canseco, and Artur W. Dubrawski. N-hits: Neural hierarchical interpolation for time series forecasting. *ArXiv*, abs/2201.12886, 2022. URL https://api.semanticscholar.org/CorpusID:246430557.

Jiahui Chen, Bingke Zhu, Hongyu Pan, and Yingying Chen. Wavetlm: Reliable time-series language modeling through task compilation. *arXiv preprint arXiv:2609.18812*, 2026.

Alexander D'Amour, Katherine A. Heller, Dan I. Moldovan, Ben Adlam, et al. Underspecification presents challenges for credibility in modern machine learning. *ArXiv*, abs/2011.03395, 2020. URL https://api.semanticscholar.org/CorpusID:226278105.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2023.

Angus Dempster, Franccois Petitjean, and Geoffrey I. Webb. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34:1454 – 1495, 2019. URL https://api.semanticscholar.org/CorpusID:204949593.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The llama 3 herd of models. 2024. URL https://api.semanticscholar.org/CorpusID:271571434.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 7:1–26, 1979. URL https://api.semanticscholar.org/CorpusID:227312712.

Elizabeth Fons, Rachneet Kaur, Soham Palande, Zhen Zeng, Tucker Balch, Manuela Veloso, and Svitlana Vyetrenko. Evaluating large language models on time series feature understanding: A comprehensive taxonomy and benchmark. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21598–21634, 2024.

Matt Gardner, Yoav Artzi, Jonathan Berant, Ben Bogin, Sihao Chen, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models' local decision boundaries via contrast sets. In *Findings*, 2020. URL https://api.semanticscholar.org/CorpusID:222124366.

Rakshitha Godahewa, C. Bergmeir, Geoffrey I. Webb, Rob J Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. *ArXiv*, abs/2105.06643, 2021. URL https://api.semanticscholar.org/CorpusID:234681550.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in neural information processing systems*, 36:19622–19635, 2023.

Hansika Hewamalage, Klaus Ackermann, and C. Bergmeir. Forecast evaluation for data scientists: common pitfalls and best practices. *Data Mining and Knowledge Discovery*, 37:788 – 832, 2022. URL https://api.semanticscholar.org/CorpusID:247594079.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. *ArXiv*, abs/1909.03368, 2019. URL https://api.semanticscholar.org/CorpusID:202538609.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, et al. Mistral 7b. *ArXiv*, abs/2310.06825, 2023. URL https://api.semanticscholar.org/CorpusID:263830494.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. In *International conference on learning representations*, volume 2024, pp. 23857–23880, 2024.

Divyansh Kaushik, Eduard H. Hovy, and Zachary Chase Lipton. Learning the difference that makes a difference with counterfactually-augmented data. *ArXiv*, abs/1909.12434, 2019. URL https://api.semanticscholar.org/CorpusID:203591519.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jangho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022. URL https://api.semanticscholar.org/CorpusID:251647808.

Yaxuan Kong, Yiyuan Yang, Yoontae Hwang, Wenjie Du, Stefan Zohren, Zhangyang Wang, Ming Jin, and Qingsong Wen. Time-mqa: Time series multi-task question answering with context enhancement. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29736–29753, 2025.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lin-Tao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *ArXiv*, abs/2310.06625, 2023. URL https://api.semanticscholar.org/CorpusID:263830644.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Autotimes: Autoregressive time series forecasters via large language models. *Advances in Neural Information Processing Systems*, 37:122154–122184, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. URL https://api.semanticscholar.org/CorpusID:53592270.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *ArXiv*, abs/1902.01007, 2019. URL https://api.semanticscholar.org/CorpusID:59599752.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *ArXiv*, abs/2211.14730, 2022. URL https://api.semanticscholar.org/CorpusID:254044221.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, et al. 2 olmo 2 furious. *ArXiv*, abs/2501.00656, 2024. URL https://api.semanticscholar.org/CorpusID:275213098.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *ArXiv*, abs/1905.10437, 2019. URL https://api.semanticscholar.org/CorpusID:166228758.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. Information-theoretic probing for linguistic structure. In *Annual Meeting of the Association for Computational Linguistics*, 2020. URL https://api.semanticscholar.org/CorpusID:215238965.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging nlp models. In *Annual Meeting of the Association for Computational Linguistics*, 2018. URL https://api.semanticscholar.org/CorpusID:21740766.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4902–4912, 2020.

Gemma Team Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, L'eonard Hussenot, et al. Gemma 2: Improving open language models at a practical size. *ArXiv*, abs/2408.00118, 2024. URL https://api.semanticscholar.org/CorpusID:270843326.

Alexis Roger, Prateek Humane, Zhenghan Tai, G. L. Legate, Andrei Mircea, Vasilii Feofanov, and Irina Rish. Llm pretraining shapes a generalizable manifold: Insights into cross-modal transfer to time series. *ArXiv*, abs/2605.20449, 2026. URL https://api.semanticscholar.org/CorpusID:288649426.

Daniel Schumacher, Erfan Nourbakhsh, Rocky Slavin, and Anthony Rios. Prompting underestimates llm capability for time series classification. *ArXiv*, abs/2601.03464, 2026. URL https://api.semanticscholar.org/CorpusID:284532767.

Zhihong Shao, Damai Dai, Daya Guo, Zihan Wang, and Huajian Xin. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *ArXiv*, abs/2405.04434, 2024. URL https://api.semanticscholar.org/CorpusID:269613809.

Mingtian Tan, Mike A Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. Are language models actually useful for time series forecasting? *Advances in Neural Information Processing Systems*, 37:60162–60191, 2024.

Elena Voita and Ivan Titov. Information-theoretic probing with minimum description length. In *Conference on Empirical Methods in Natural Language Processing*, 2020. URL https://api.semanticscholar.org/CorpusID:214693050.

Patrick Wagner, Nils Strodthoff, Ralf Bousseljot, Dieter Kreiseler, Fatima I. Lunze, Wojciech Samek, and Tobias Schaeffter. Ptb-xl, a large publicly available electrocardiography dataset. *Scientific Data*, 7, 2020. URL https://api.semanticscholar.org/CorpusID:218865062.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*, 2024.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Neural Information Processing Systems*, 2021. URL https://api.semanticscholar.org/CorpusID:235623791.

Haixu Wu, Teng Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *ArXiv*, abs/2210.02186, 2022. URL https://api.semanticscholar.org/CorpusID:252715491.

Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. Chatts: Aligning time series with llms via synthetic data for enhanced understanding and reasoning. *arXiv preprint arXiv:2412.03104*, 2024.

An Yang, An-Feng Li, Baosong Yang, et al. Qwen3 technical report. 2025. URL https://api.semanticscholar.org/CorpusID:278602855.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yu Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *AAAI Conference on Artificial Intelligence*, 2021. URL https://api.semanticscholar.org/CorpusID:237497421.

Ailing Zeng, Mu-Hwa Chen, L. Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *AAAI Conference on Artificial Intelligence*, 2022. URL https://api.semanticscholar.org/CorpusID:249097444.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wan Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. *ArXiv*, abs/2012.07436, 2020. URL https://api.semanticscholar.org/CorpusID:229156802.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, 2022. URL https://api.semanticscholar.org/CorpusID:246430171.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

A Unified Experimental Protocol

The following sections provide the complete settings and results for the three branches in Figure 1. Table 2 summarizes which models and samples enter each comparison. QA is shorthand for question answering. In all detailed tables, P means pretrained weights and R means random initialization. The letters A, B, and C identify the task-specific readout, question-conditioned scorer, and native answer, respectively; they are distinct from the question-type identifiers defined in Appendix D.

Table 2: **Experimental coverage.** Learned readouts compare pretrained and random backbones. Real-data questions use the same 42 test source groups across models and seeds. Native answers use pretrained models only.

Comparison	Models	Supervision and readout	Input and test units	Measures
Forecasting	10	Process family or future expert; affine forecast head	64-value context; 120 routing and 300 forecast windows per seed	Routing accuracy; future error
Expected risk	5	Linear router; mean loss over 20 futures	Control's own contexts and reference fits	Routing accuracy
Learned question readouts	10	Task-specific classifier or question-conditioned scorer	Real observations; 42 groups	Item and paired accuracy
Native answers	5	Free generation or constrained choice	Real observations; 42 groups	Accuracy and validity

A.1 Model Coverage and Attribution

The ten forecasting and learned question-answering backbones are Qwen3-8B base, Llama-3.1-8B, Llama-3.2-3B, Gemma-2-9B, Gemma-2-2B, Mistral-7B-v0.3, DeepSeek-LLM-7B, DeepSeek-V2-Lite, OLMo-2-7B, and OLMo-2-13B. The expected-risk subset contains Qwen3-8B base, Llama-3.1-8B, Gemma-2-9B, DeepSeek-LLM-7B, and OLMo-2-7B. The native-question-answering subset contains Qwen3-8B base, Llama-3.1-8B, Gemma-2-9B, DeepSeek-V2-Lite, and OLMo-2-7B. These five-model subsets have different membership. All checkpoints are base models.

The model-family reports describe Qwen3 (Yang et al., 2025), Llama 3 (Dubey et al., 2024), Gemma 2 (Riviere et al., 2024), Mistral 7B (Jiang et al., 2023), DeepSeek LLM and DeepSeek-V2 (Bi et al., 2024; Shao et al., 2024), and OLMo 2 (OLMo et al., 2024). These references document the families and their training; the model IDs identify the particular checkpoints used here.

Qwen3-8B uses the base checkpoint in both branches. Their feature extraction protocols differ, as specified below.

Within each architecture, the random condition uses the model's native initialization while retaining its tokenizer and special-token handling. Seeds 7, 17, and 27 identify the repeated fits and random controls. Pretrained feature caches can be shared across seeds. In particular, the question-conditioned scorer pairs one pretrained temporal cache with the seed-specific random cache and the same head initialization. The three seeds therefore index repeated fits under the specified initialization conditions, with pretrained weights held fixed.

A.2 Representation and Optimization

MLP denotes a multilayer perceptron and MSE mean squared error. BA denotes balanced accuracy, which weights the accuracy of each class equally. A random projection reduces feature width with a fixed matrix; it is distinct from random initialization of a language-model backbone. AR denotes an autoregressive model, with the number of lags specified by its order.

Forecasting serializes context-normalized values as signed decimals with two places, clipped to $[-9.99, 9.99]$, within a history/forecast prompt. The representation is the final nonpadding token's last-layer state. QA serializes timestamps, channel names, units and numerical arrays to four decimal

places and mean-pools evidence token states at the last layer. The QA temporal branch contains no question, candidate, or gold. These extraction rules are specific to the forecasting and QA experiments, respectively.

The synthetic linear classification protocol uses full-batch optimization for 2,000 steps, learning rate 0.5 and ℓ_2 coefficient 0.001. The affine numerical head predicts a 16-step vector and trains for 300 epochs with AdamW (Loshchilov & Hutter, 2017), learning rate 10^{-3} and weight decay 10^{-2}. The MLP variants change the readout family while retaining the frozen backbone. Their seed-level scores and parameter counts are provided in the supplementary data.

QA A uses a layer-normalized linear semantic classifier for each task. B scores each question-candidate pair with a shared MLP after fixed projections of both inputs to 128 dimensions. Its hidden width is 64 with GELU activation. The A/B budget is 100 epochs, AdamW learning rate 10^{-3}, weight decay 10^{-2} and batch size 256. P/R conditions share head initialization; B also pairs minibatch order. Its question branch is pretrained in both conditions, so the contrast concerns the temporal representation. Train/validation/test membership is fixed before fitting. Tables in Appendix E report seed-level results on the same test groups.

Native protocols perform no fitting. Free generation uses deterministic decoding followed by the strict option-letter parser; a legal letter may optionally follow "Answer:" and precede a period. Extra explanations are invalid. Constrained choice instead scores the allowed answer continuations by their summed token log likelihood and chooses the largest. It produces legal answers by construction. Both are independent item calls with no prior answers in context.

B Forecasting Tasks and Complete Primary Results

B.1 Synthetic Processes and Normalization

Each sample contains a context $x_{1:64}$ and future $y_{1:16}$. Trend uses slope $a \sim U(0.03, 0.18)$. Periodic signals use amplitude $U(0.5, 1.5)$, period 12 or 24, and phase $U(0, 2\pi)$. Local signals follow an AR(1) process with coefficient $U(0.7, 0.95)$. Additive noise has standard deviation 0.05 for trend and periodic processes and 0.15 for local processes. Mixtures combine a weaker slope $U(0.01, 0.06)$ with amplitude $U(0.4, 1.2)$; regime-change sequences switch from slope 0.08 in the initial 60% to a period-12 sinusoid of amplitude 1. Context statistics alone normalize both context and future.

Recognition and numerical readout use 90 training windows and 60 test windows from each of the five process families, for totals of 450 and 300 per seed. Routing trains on 270 primitive windows and tests on 120 unseen-composition windows, balanced across the three future-defined expert classes. A local component of the out-of-distribution test follows $u_t = 0.9u_{t-1} + \epsilon_t$, with innovation standard deviation 0.3, plus a weak amplitude-0.15 period-24 sinusoid. The richer-local control changes the innovation standard deviation to 0.28 and the sinusoid amplitude to 0.4.

B.2 Expert Bank and Label Targets

The fixed expert bank contains a least-squares trend extrapolator, a seasonal predictor and a damped AR(5) predictor. Each produces the same 16-step horizon. Process-family training maps trend, periodic and local classes to the corresponding expert. Single-future labels minimize realized normalized MSE over this bank. Both trained routers are evaluated against the same realized-future winner; their P/R effects are therefore comparable under a common evaluation, even though the supervised targets differ.

The trend expert fits a degree-one polynomial in time. The periodic expert detrends the context, evaluates autocorrelation at lags 3-31, accepts a seasonal lag at correlation at least 0.4, continues the most recent seasonal pattern, and rolls forward an AR(1) residual correction with coefficient clipped to $[0, 0.95]$. The local expert fits AR(5) by least squares. If its recurrence is unstable, the coefficient vector is multiplied by 0.9 repeatedly, for at most 60 iterations, before recursive prediction.

The expected-risk control holds process parameters and context fixed, draws 20 future-noise realizations, and selects the expert with minimum mean squared error across those realizations. Table 4 reports this control's own reference fits so that its five-model contrasts remain separate from the ten-model experiment.

Table 3: Ten-model mean forecasting results. Routing columns are balanced accuracy in percent; the last two columns are normalized future MSE. Three seeds per cell.

Model	Family P	R	Future P	R	MSE P	R
Qwen3-8B	83.33	52.22	26.11	45.56	0.58767	0.62533
Llama-3.1-8B	74.44	51.67	31.11	48.33	0.56020	0.88487
Llama-3.2-3B	66.39	51.67	39.44	51.67	0.57383	0.70543
Gemma-2-9B	64.44	44.44	36.94	42.50	0.60180	1.94137
Gemma-2-2B	59.72	44.72	33.34	46.67	0.64043	1.56957
Mistral-7B-v0.3	63.89	53.33	28.06	41.11	0.52410	0.69657
DeepSeek-LLM-7B	60.83	53.33	31.67	53.06	0.62700	0.69487
DeepSeek-V2-Lite	79.72	48.33	25.28	40.28	0.64530	0.58273
OLMo-2-7B	58.61	49.17	33.61	45.83	0.50413	0.58507
OLMo-2-13B	58.61	50.28	29.17	46.39	0.55397	0.58403

Table 4: Five-model expected-risk experiment, using its own reference fits. BA and P-R effects use percent and percentage points, respectively.

Model	Target	P, %	R, %	Δ, pp
Qwen3-8B	Family	83.33	51.94	31.39
	Single future	26.11	45.28	-19.16
	Expected risk	31.66	51.39	-19.73
Llama-3.1-8B	Family	74.44	51.67	22.78
	Single future	32.50	48.33	-15.83
	Expected risk	38.61	50.28	-11.67
Gemma-2-9B	Family	64.44	44.44	20.00
	Single future	36.94	42.50	-5.56
	Expected risk	37.78	47.78	-10.00
DeepSeek-LLM-7B	Family	60.83	45.83	15.00
	Single future	31.67	50.00	-18.34
	Expected risk	35.00	48.61	-13.61
OLMo-2-7B	Family	58.61	49.17	9.44
	Single future	33.61	45.83	-12.22
	Expected risk	33.33	44.44	-11.11

B.3 Aggregation Across Seeds

For each model, results are aggregated over three seeds. Figure 2 reports the mean of the paired pretrained-random differences in balanced routing accuracy. Figure 3 first averages the absolute MSE across seeds within each weight condition, then computes the relative error reduction from these means. The reported reduction is therefore based on the ratio of mean MSEs, rather than the mean of per-seed relative reductions.

C Forecasting Extension Controls

C.1 Width, Readout and Local Processes

Table 5 reports family routing after a fixed random projection reduces the feature width. Each entry is a mean over three seeds. All ten models have positive mean effects at each tested width. The projection controls readout width within each architecture.

Table 5: Mean P-R family-routing balanced-accuracy differences, in percentage points, at fixed random-projection widths. Each cell averages three seeds.

Model	64	128	256	384	512	768	1024	2048
Qwen3-8B	31.39	28.61	31.67	35.55	38.61	38.05	33.61	31.39
Llama-3.1-8B	28.89	29.72	24.44	28.62	30.56	29.44	30.55	27.50
Llama-3.2-3B	16.39	17.50	24.72	16.11	16.95	21.94	17.50	15.00
Gemma-2-9B	19.44	25.84	21.67	20.00	24.17	27.78	27.22	19.72
Gemma-2-2B	21.67	22.22	19.72	25.00	27.50	19.17	19.72	14.73
Mistral-7B-v0.3	24.45	16.66	15.83	21.67	36.39	28.06	23.61	11.11
DeepSeek-LLM-7B	17.50	14.17	15.83	14.44	16.39	18.89	16.66	14.73
DeepSeek-V2-Lite	32.50	36.39	38.61	36.94	37.50	35.56	34.17	–
OLMo-2-7B	6.94	4.44	6.11	9.45	11.11	11.94	8.05	7.78
OLMo-2-13B	23.05	15.28	20.28	10.28	20.00	14.17	13.06	10.55

Table 6: Additional forecasting controls. Recognition uses accuracy; MLP routing and richer-local use BA. Their effects are P-R in pp. MLP numerical error reduction is R-P normalized MSE. Dashes denote unavailable results.

Model	Recognition	Shuffled	MLP MSE	MLP router	Richer local
Qwen3-8B	6.22	-0.67	–	–	28.89
Llama-3.1-8B	4.78	-6.33	0.20440	29.45	19.72
Llama-3.2-3B	3.11	-3.56	0.10963	17.78	12.78
Gemma-2-9B	17.78	31.89	0.41917	18.06	19.73
Gemma-2-2B	12.44	24.78	0.22823	18.33	11.66
Mistral-7B-v0.3	7.67	8.55	0.08120	24.44	10.83
DeepSeek-LLM-7B	5.11	7.11	0.04647	12.50	6.39
DeepSeek-V2-Lite	4.34	1.77	0.10540	36.11	28.89
OLMo-2-7B	4.89	1.66	0.13413	11.66	10.00
OLMo-2-13B	4.67	7.56	0.09750	18.61	7.22

Table 6 includes process recognition, recognition after time shuffling, nonlinear readouts, and the richer-local test. Recognition measures accuracy across five families, while routing measures balanced accuracy across three experts. MLP denotes a multilayer perceptron. Its numerical head is evaluated by mean squared error, so a reduction is favorable. DeepSeek-V2-Lite improves under this nonlinear numerical head, whereas its error increases with the affine head in the main text. This contrast shows how the outcome changes with the readout.

C.2 Time-Series Foundation Models

Six additional backbones use their native numerical input interfaces: Chronos-T5-small/base, Chronos-Bolt-small, TimesFM-2.5-200M, MOMENT-1-large and Moirai-1.1-R-small. The P/R contrasts retain each model's architecture and extraction conventions. Table 7 gives both absolute conditions. Family-routing effects range from -6.39 to $+11.11$ points. Their mixed signs place the LM pattern in a broader empirical context, with architecture and numerical interface varying alongside pretraining domain.

Table 7: Six time-series foundation models. Family/future entries are routing balanced accuracy in percent; MSE entries are normalized numerical-readout error. Their numerical input interfaces differ from LM serializations.

Model	Task	P	R
chronos_bolt-small	Family	68.33	65.56
	Future	39.44	34.45
	MSE	0.40540	1.90653

Model	Task	P	R
chronos_t5-base	Family	51.11	40.00
	Future	26.94	36.11
	MSE	1.08253	1.90680
chronos_t5-small	Family	41.11	41.11
	Future	28.89	34.44
	MSE	0.82820	1.90680
moirai-1.1-R-small	Family	56.39	62.78
	Future	23.05	38.61
	MSE	0.39373	0.45133
moment-1-large	Family	63.06	66.11
	Future	38.05	45.00
	MSE	0.53527	0.65190
timesfm_2.5-200m	Family	65.28	70.28
	Future	35.84	33.33
	MSE	0.40347	0.50827

D Construction of Real Time-Series Evidence Families

D.1 Sources, Groups and Units

Table 8: Source-group split counts and visible window lengths. Each selected source group contributes one family.

Domain	Train	Validation	Test	Samples/window
biomedical	52	7	7	96
mobility	49	8	9	48
energy	46	7	13	48
environment	46	7	13	32

PTB-XL 1.0.3 provides de-identified ECG observations at 100 Hz, with lead voltages in mV (Wagner et al., 2020). Grouping uses patient ID and respects its official folds, keeping records of the same patient together. The biomedical windows have 96 samples. The City of Melbourne pedestrian data, accessed through the Monash archive (Godahewa et al., 2021), supply hourly observations grouped by sensor. The electricity data are the Monash hourly derivative of the UCI client series, grouped by client. These two domains use 48 observations per window. Australian weather data use observation series, rather than forecasts, grouped by station; the 32 daily observations retain temperature in degrees Celsius and rain in mm. Windows with missing-value substitutions are excluded.

Each domain contributes 66 selected groups. Patient-respecting folds and fixed group hashing yield Table 8; all questions, edited variants and option orders inherit this split. One selected family per group makes the 42 test families 42 source groups. Their domain counts differ, so domain-macro and pooled estimates are distinct. The reconstruction manifest retains record IDs, source/member hashes, sample boundaries, channels, timestamps, units and edit parameters.

PTB-XL is distributed under CC BY 4.0; the plotted excerpt carries attribution in the artifact. The Australian source is recorded under CC BY-SA 3.0 Australia. The Melbourne archive attribution/record metadata and the exact hourly electricity transformation require their upstream release documentation. Accordingly, the artifact redistributes their reconstruction metadata rather than raw observations. Electricity questions use "reported load value": the hourly derivative is not relabeled as energy in kWh.

D.2 Question Semantics and Visible Golds

The six original-observation questions ask for the sign of the fitted trend, the half with the higher target mean, the quarter containing the target maximum, the control statistic's half comparison, the

joint target and control description, and the peak location within the higher-mean half. We denote these question types A through F in that order. The peak-quarter and joint-description questions have four options; all others have two. Every original-edited pair keeps the same question wording and option mapping.

Gold computation applies the semantic functions to the visible four-decimal arrays, including after shuffling. It does not rely on hidden higher-precision values. A least-squares slope defines A; half means define B; an argmax and quarter boundaries define C; D uses the recorded untouched-channel mean or within-half range; E combines the explicit B/D predicates; F selects the higher-mean half and localizes its peak. The E question states both statistics explicitly within the item, so it can be interpreted without any preceding answer.

D.3 Edit and Control Invariance

Let μ_1, μ_2 be the target-channel means in the two halves and let L denote the lower-mean half. The primary edit adds a constant $\delta > |\mu_1 - \mu_2|$ to every value in L, leaving other target samples and other channels unchanged. Golds are checked after serialization. If the control uses another channel, its data are unchanged. If it compares within-half ranges of the same channel, a constant offset preserves those ranges. An edit is admitted only when the target answer changes and the control answer remains the same. All 42 test families pass both predicates.

This construction tests whether answers follow the explicitly supplied numbers. The offset is a derived numerical edit of a recorded signal. The post-hoc equal-gap variant sets $\delta = 2|\mu_1 - \mu_2|$ before visible-value validation. It reverses the half-mean relation while preserving the magnitude of the gap.

D.4 Example and Independent Calls

The question example in Figure 1 comes from the first biomedical test family in lexicographic order, `ptbxl_ecg00129_0240`. The example is selected independently of model predictions. The public record is ECG 129, patient group 16463, official fold 10, samples 240-335 at 100 Hz. Lead V1 is the target and lead II the control. In the associated evidence family, adding 0.1612 mV to the second half of V1 changes the higher-mean half from first to second; the control answer remains second. The figure shows the two channels' half means before and after the edit; its shared-input waveform is schematic. The original and edited arrays and their source hash are supplied with the plotting data.

Native prompts concatenate the visible evidence with one self-contained question and its candidates, ending with an instruction to return one option letter. Original target, original control, edited target and edited control are four separate calls. No answer history or feedback links them. "Retaining the old answer" therefore denotes the same semantic prediction across the two independently evaluated inputs.

E Complete QA Results

E.1 Metrics and Complete Readout Results

All accuracies count invalid outputs as incorrect. VOR denotes valid-output rate, the fraction of outputs that the parser accepts. The table column Obs. acc. gives accuracy on the six original-observation questions, while All acc. includes the two edited items. Within each domain, all-item scoring first averages repeated variants within a family and task, then families within groups, groups within tasks, and tasks. The final score weights the four domains equally. Repeating the target and control questions after an edit therefore does not double the weight assigned to those tasks.

TUS denotes target update success and requires both target answers to be correct. CIS denotes control invariance success and requires both control answers to be correct. FCJS denotes four-cell joint success and requires all four answers. QSJA denotes question-switch joint accuracy and equals $\frac{1}{2}\langle C_{t0}C_{c0} + C_{t1}C_{c1}\rangle$. It measures whether both questions are answered correctly for each evidence condition. Each metric averages groups within domains and then weights the domains equally. Predicted option letters are mapped back to their semantic answers before scoring. In the table labels,

A is the task-specific readout and B the question-conditioned scorer; P and R identify pretrained and random temporal weights.

Table 9: Complete A/B model means, in percent, averaging seeds 7/17/27; all conditions use the same test groups.

Model	Fit	Obs. acc.	All acc.	TUS	CIS	FCJS	QSJA	VOR
Qwen3-8B	A P	45.98	45.74	15.79	41.76	7.91	23.00	100.00
	A R	48.47	49.65	10.09	51.55	5.96	26.56	100.00
	B P	47.68	45.84	0.00	45.51	0.00	22.76	100.00
	B R	50.20	48.30	1.28	58.19	0.64	29.05	100.00
Llama-3.1-8B	A P	42.91	46.13	34.77	35.04	9.36	21.70	100.00
	A R	49.43	51.30	28.32	46.85	19.33	32.72	100.00
	B P	48.63	47.57	0.93	44.54	0.93	22.73	100.00
	B R	44.23	44.05	4.04	38.44	0.00	19.67	100.00
Llama-3.2-3B	A P	41.72	44.05	34.13	35.39	9.81	21.87	100.00
	A R	46.12	47.67	10.72	42.53	6.59	23.92	100.00
	B P	48.62	47.08	0.00	47.10	0.00	23.55	100.00
	B R	45.50	45.62	0.64	43.21	0.00	21.29	100.00
Gemma-2-9B	A P	48.93	52.92	21.16	48.25	10.00	32.40	100.00
	A R	46.10	47.26	3.95	47.29	2.38	24.56	100.00
	B P	50.07	48.90	0.64	46.10	0.00	23.05	100.00
	B R	49.53	48.45	1.19	52.30	1.19	27.39	100.00
Gemma-2-2B	A P	56.47	56.55	20.09	47.95	8.76	32.36	100.00
	A R	47.81	48.15	9.10	51.60	5.32	26.77	100.00
	B P	49.00	48.39	0.00	47.74	0.00	23.87	100.00
	B R	50.79	50.41	0.64	55.72	0.00	27.86	100.00
Mistral-7B-v0.3	A P	44.98	48.87	33.39	40.72	9.10	26.93	100.00
	A R	48.50	50.55	17.99	48.17	9.63	29.17	100.00
	B P	50.24	48.46	0.64	46.17	0.64	23.41	100.00
	B R	52.19	50.42	1.83	52.49	0.00	26.42	100.00
DeepSeek-LLM-7B	A P	45.75	48.92	21.24	43.19	7.89	26.51	100.00
	A R	48.43	50.19	19.56	48.15	9.63	29.62	100.00
	B P	48.96	47.71	0.00	47.10	0.00	23.55	100.00
	B R	52.26	51.33	2.76	51.74	1.19	26.79	100.00
DeepSeek-V2-Lite	A P	50.40	49.40	20.07	42.38	6.89	24.18	100.00
	A R	48.72	49.49	3.40	50.91	1.83	26.05	100.00
	B P	47.21	45.98	0.00	48.01	0.00	24.00	100.00
	B R	49.46	48.28	0.64	52.56	0.00	25.69	100.00
OLMo-2-7B	A P	44.95	44.77	21.99	32.00	4.13	20.27	100.00
	A R	45.62	45.37	1.92	38.50	0.64	18.01	100.00
	B P	48.75	46.96	0.00	47.65	0.00	23.82	100.00
	B R	45.50	46.23	0.64	41.17	0.64	20.58	100.00
OLMo-2-13B	A P	43.54	45.50	34.77	30.33	10.16	23.15	100.00
	A R	48.66	48.19	4.32	47.42	3.40	24.49	100.00
	B P	48.24	46.36	0.00	46.46	0.00	23.23	100.00
	B R	46.15	45.67	0.00	40.90	0.00	20.45	100.00

Table 10: A/B seed-level paired scores, in percent. Scores describe repeated fits on the same 42 test source groups.

Model	Fit	Seed	TUS	CIS	FCJS	QSJA
Qwen3-8B	A P	7	27.29	45.18	14.10	25.11
		17	20.09	40.48	9.62	22.70
		27	0.00	39.62	0.00	21.20
	A R	7	10.47	50.34	5.77	24.21
		17	14.04	51.19	10.20	29.73
		27	5.77	53.11	1.92	25.73
	B P	7	0.00	44.32	0.00	22.16
		17	0.00	44.32	0.00	22.16
		27	0.00	47.89	0.00	23.95
	B R	7	1.92	57.81	0.00	27.95
		17	1.92	57.02	1.92	31.26
		27	0.00	59.74	0.00	27.95
Llama-3.1-8B	A P	7	36.69	35.50	12.12	25.87
		17	32.84	37.70	5.77	15.28
		27	34.77	31.93	10.20	23.95
	A R	7	23.90	50.06	18.13	34.10
		17	28.88	43.71	19.54	30.66
		27	32.17	46.76	20.33	33.41
	B P	7	0.00	47.10	0.00	23.55
		17	2.78	43.25	2.78	23.02
		27	0.00	43.25	0.00	21.63
	B R	7	8.27	33.00	0.00	17.46
		17	1.92	33.85	0.00	18.71
		27	1.92	48.47	0.00	22.85
Llama-3.2-3B	A P	7	34.77	31.35	15.96	25.31
		17	32.84	41.54	5.77	19.38
		27	34.77	33.27	7.69	20.91
	A R	7	7.42	41.51	3.57	22.54
		17	10.20	46.21	3.57	23.93
		27	14.56	39.87	12.64	25.29
	B P	7	0.00	47.10	0.00	23.55
		17	0.00	47.10	0.00	23.55
		27	0.00	47.10	0.00	23.55
	B R	7	0.00	47.10	0.00	23.55
		17	1.92	39.62	0.00	18.85
		27	0.00	42.92	0.00	21.46
Gemma-2-9B	A P	7	24.30	47.04	3.85	29.18
		17	16.03	45.39	7.69	27.93
		27	23.17	52.32	18.47	40.09
	A R	7	5.49	43.71	3.57	24.60
		17	2.78	48.69	0.00	22.56
		27	3.57	49.48	3.57	26.53
	B P	7	0.00	49.88	0.00	24.94
		17	0.00	41.33	0.00	20.67
		27	1.92	47.10	0.00	23.55
	B R	7	0.00	55.31	0.00	28.62
		17	0.00	54.24	0.00	28.08
		27	3.57	47.34	3.57	25.46

Model	Fit	Seed	TUS	CIS	FCJS	QSJA
Gemma-2-2B	A P	7	25.43	47.95	12.18	35.30
		17	22.65	47.95	9.40	31.17
		27	12.18	47.95	4.70	30.60
	A R	7	8.55	50.34	4.70	25.34
		17	12.12	49.21	7.42	27.35
		27	6.62	55.25	3.85	27.63
	B P	7	0.00	49.02	0.00	24.51
		17	0.00	47.10	0.00	23.55
		27	0.00	47.10	0.00	23.55
	B R	7	0.00	54.52	0.00	27.26
		17	0.00	50.40	0.00	25.20
		27	1.92	62.24	0.00	31.12
Mistral-7B-v0.3	A P	7	36.42	41.00	14.90	29.87
		17	32.84	43.47	3.85	23.66
		27	30.92	37.70	8.55	27.26
	A R	7	19.54	47.62	12.12	29.87
		17	12.12	49.27	7.42	27.38
		27	22.31	47.62	9.34	30.27
	B P	7	0.00	44.32	0.00	22.16
		17	1.92	47.10	1.92	24.51
		27	0.00	47.10	0.00	23.55
	B R	7	1.92	55.04	0.00	28.05
		17	3.57	50.40	0.00	25.20
		27	0.00	52.05	0.00	26.02
DeepSeek-LLM-7B	A P	7	18.74	34.92	5.77	25.05
		17	23.44	46.25	5.77	22.83
		27	21.52	48.41	12.12	31.65
	A R	7	23.11	51.19	12.91	32.88
		17	17.67	44.84	6.62	26.16
		27	17.89	48.41	9.34	29.84
	B P	7	0.00	46.25	0.00	23.12
		17	0.00	47.10	0.00	23.55
		27	0.00	47.95	0.00	23.98
	B R	7	3.57	47.34	3.57	24.07
		17	0.00	55.04	0.00	27.52
		27	4.70	52.84	0.00	28.77
DeepSeek-V2-Lite	A P	7	30.07	45.97	16.82	29.61
		17	27.35	41.54	3.85	20.77
		27	2.78	39.62	0.00	22.16
	A R	7	0.00	46.49	0.00	23.24
		17	7.42	51.19	5.49	27.38
		27	2.78	55.04	0.00	27.52
	B P	7	0.00	44.32	0.00	22.16
		17	0.00	44.32	0.00	22.16
		27	0.00	55.37	0.00	27.69
	B R	7	0.00	61.66	0.00	30.83
		17	0.00	50.61	0.00	25.31
		27	1.92	45.42	0.00	20.92
OLMo-2-7B	A P	7	26.50	31.07	6.62	21.60
		17	32.84	31.07	3.85	18.03
		27	6.62	33.85	1.92	21.20
	A R	7	1.92	36.29	0.00	16.36
		17	1.92	39.07	0.00	17.61
		27	1.92	40.14	1.92	20.07
	B P	7	0.00	49.02	0.00	24.51
		17	0.00	47.10	0.00	23.55
		27	0.00	46.83	0.00	23.41
	B R	7	0.00	41.54	0.00	20.77
		17	1.92	43.47	1.92	21.73
		27	0.00	38.49	0.00	19.25

Model	Fit	Seed	TUS	CIS	FCJS	QSJA
OLMo-2-13B	A P	7	36.69	26.37	11.26	21.99
		17	34.77	33.27	11.54	23.79
		27	32.84	31.35	7.69	23.66
	A R	7	1.92	48.41	1.92	23.38
		17	11.05	45.42	8.27	25.89
		27	0.00	48.41	0.00	24.21
	B P	7	0.00	47.10	0.00	23.55
		17	0.00	49.02	0.00	24.51
		27	0.00	43.25	0.00	21.63
	B R	7	0.00	35.99	0.00	17.99
		17	0.00	49.02	0.00	24.51
		27	0.00	37.70	0.00	18.85

Table 11: Domain-level mean scores, in percent. Each row averages the seeds and models in the stated interface.

Fit	Domain	Obs. acc.	All acc.	TUS	CIS	FCJS
A P	biomedical	35.48	36.03	10.48	31.43	4.29
	energy	53.68	55.51	32.56	40.77	16.92
	environment	47.35	49.47	17.69	61.79	4.62
	mobility	49.75	52.13	42.22	24.81	7.78
A R	biomedical	51.59	53.53	15.71	69.52	12.38
	energy	56.71	57.91	16.15	48.46	9.23
	environment	45.68	46.00	4.10	48.97	2.05
	mobility	37.16	37.69	7.78	22.22	2.22
B P	biomedical	35.40	29.60	0.00	30.00	0.00
	energy	54.70	54.36	0.51	47.69	0.26
	environment	44.74	44.66	0.00	65.90	0.00
	mobility	60.12	60.68	0.37	42.96	0.37
B R	biomedical	47.86	41.39	1.90	53.81	0.95
	energy	53.38	54.83	1.03	49.49	0.00
	environment	44.02	44.42	1.79	57.69	0.51
	mobility	49.07	50.86	0.74	33.70	0.00
Free	biomedical	13.33	14.29	0.00	20.00	0.00
	energy	17.95	18.08	4.62	23.08	3.08
	environment	13.59	13.08	3.08	12.31	1.54
	mobility	22.96	21.67	2.22	33.33	2.22
Constrained	biomedical	38.57	39.76	5.71	40.00	2.86
	energy	44.36	44.62	6.15	40.00	1.54
	environment	36.41	35.38	9.23	35.38	7.69
	mobility	45.93	46.67	0.00	55.56	0.00

The tables give model means, individual seed outcomes, and domain averages for both learned readouts. The accompanying data also contain every domain-by-model-by-seed score. All fits reuse the same 42 test source groups.

E.2 Native Modes, Missing Evidence and Conditional Counts

Table 12: Native counts on the 168 core items. Initial, update and stale count target pairs; update/stale are conditional on an initially correct target. Joint uses all 42 families.

Model	Mode	Correct	Valid	Initial	Update	Stale	Joint
Qwen3-8B	Free	81/168	168/168	20	2	18	1/42
	Constrained	80/168	168/168	19	1	18	1/42
Llama-3.1-8B	Free	48/168	91/168	11	2	8	2/42
	Constrained	80/168	168/168	18	3	15	3/42
Gemma-2-9B	Free	0/168	0/168	0	0	0	0/42
	Constrained	84/168	168/168	21	0	21	0/42
DeepSeek-V2-Lite	Free	0/168	0/168	0	0	0	0/42
	Constrained	83/168	168/168	18	6	12	1/42
OLMo-2-7B	Free	81/168	168/168	24	2	22	1/42
	Constrained	80/168	168/168	24	2	22	2/42

Table 13: Native results, in percent. Four-cell metrics use full-evidence families satisfying both edit predicates; shuffled-input rows report item accuracy and validity. Native conditions have one deterministic evaluation each.

Model	Mode	Input	Obs. acc.	All acc.	TUS	CIS	FCJS	QSJA	VOR
Qwen3-8B	Free	full	34.15	35.24	3.85	44.69	1.92	24.27	80.34
		Q only	23.60	24.29	–	–	–	–	59.82
		Shuffle	33.76	33.79	–	–	–	–	79.93
	Constrained	full	46.42	47.68	1.92	44.69	1.92	24.27	100.00
		Q only	47.60	48.77	–	–	–	–	100.00
		Shuffle	42.34	43.01	–	–	–	–	100.00
Llama-3.1-8B	Free	full	16.83	16.76	4.70	28.21	4.70	15.06	51.16
		Q only	0.00	0.00	–	–	–	–	0.00
		Shuffle	19.65	18.50	–	–	–	–	50.89
	Constrained	full	39.27	40.79	7.42	43.04	7.42	26.62	100.00
		Q only	40.82	41.03	–	–	–	–	100.00
		Shuffle	43.64	44.94	–	–	–	–	100.00
Gemma-2-9B	Free	full	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Q only	0.00	0.00	–	–	–	–	0.00
		Shuffle	0.00	0.00	–	–	–	–	0.00
	Constrained	full	41.60	41.81	0.00	49.60	0.00	24.80	100.00
		Q only	41.60	41.81	–	–	–	–	100.00
		Shuffle	40.00	41.56	–	–	–	–	100.00
DeepSeek-V2-Lite	Free	full	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Q only	0.00	0.00	–	–	–	–	0.00
		Shuffle	0.00	0.00	–	–	–	–	0.00
	Constrained	full	39.29	39.39	13.19	35.56	1.92	21.49	100.00
		Q only	40.63	40.42	–	–	–	–	100.00
		Shuffle	43.16	43.56	–	–	–	–	100.00
OLMo-2-7B	Free	full	33.82	31.88	3.85	38.00	1.92	26.05	74.70
		Q only	33.21	32.28	–	–	–	–	73.39
		Shuffle	41.29	39.44	–	–	–	–	77.75
	Constrained	full	40.01	38.38	3.85	40.78	3.85	24.66	100.00
		Q only	41.84	40.90	–	–	–	–	100.00
		Shuffle	48.57	47.12	–	–	–	–	100.00

The core native count table covers four questions per family, giving 168 items. The all-item matrix covers all 336 test items. Initial counts are model-specific: across the five constrained-choice models, 100 target answers are initially correct, and 88 of those retain the old answer after editing. The other

12 update correctly. These counts condition on initial correctness and repeat families across models. Four-cell joint counts instead use all 42 families for each model.

Shuffling has its own golds recomputed from the shuffled visible values. It changes 168 of the 336 test-item golds, and 12 of 42 shuffled families retain both target-change and control-preservation predicates. The full-family paired evaluation therefore uses original/full evidence. The shuffled rows report item accuracy and validity for the altered inputs.

E.3 Semantic and Question-Only References

The train-only task-semantic majority reference reaches 48.31% domain-macro all-item accuracy, compared with 41.67% for the uniform-choice expectation and 41.81% for a global-position majority. A domain-and-task semantic reference receives privileged domain metadata and reaches 47.30%. These are post-hoc references calculated without fitting test labels. They provide item-accuracy references; paired evidence response is measured by TUS, CIS, and FCJS.

B's question-only evaluation covers all ten models, both temporal-branch conditions and all three seeds. Because both members of a target pair have identical questions/options and opposite golds, a deterministic question-only scorer cannot answer both correctly. Zero TUS is thus entailed by the paired construction. The learned positive-control question-only models in Appendix F are separately trained controls.

E.4 Answer Transitions and Half-Mean Control Subset

We analyze answer transitions within the 42 test evidence families. Table 14 separates correct-to-correct, correct-to-wrong, wrong-to-correct, and wrong-to-wrong transitions. These four outcomes partition all pairs. A semantic flip is counted when both outputs are valid and their semantic answers differ; pairs with invalid outputs are reported separately. Each score averages source groups within a domain, gives domains equal weight, then averages seeds within models and models within the stated set. The independent sampling unit is the source group.

For a changed binary target, a flip can make both answers correct or both wrong. For an unchanged binary control, a flip makes one of the answers wrong. Thus a higher flip rate is not itself evidence of more reliable updating. The task-specific readout's updating gain coexists with more target and control changes. These transition counts distinguish the observed answer changes from correct updates.

Table 14: Answer transitions on the 42 test groups, in percent. CC, CW, WC and WW describe correctness before and after editing; their rows sum to 100. Flip counts a semantic change with both outputs valid. Invalid counts pairs with at least one invalid output. Scores weight domains, seeds and models as in the main analysis. Native rows summarize five pretrained models, while learned-readout rows summarize the complete set of ten models.

Readout	Weights	Fact	CC	CW	WC	WW	Flip	Invalid
Task-specific	P	Target	25.74	20.09	41.26	12.91	38.65	0.00
		Control	39.70	6.19	5.67	48.44	11.86	0.00
	R	Target	10.94	36.33	47.64	5.09	16.03	0.00
		Control	47.30	0.62	1.26	50.82	1.88	0.00
Question-conditioned	P	Target	0.22	58.34	41.35	0.09	0.31	0.00
		Control	46.64	0.09	0.09	53.18	0.19	0.00
	R	Target	1.37	52.53	44.51	1.60	2.96	0.00
		Control	48.67	1.14	0.71	49.48	1.84	0.00
Native choice	P	Target	5.27	40.89	47.89	5.95	11.22	0.00
		Control	42.74	6.72	3.21	47.34	9.92	0.00
Native free	P	Target	2.48	22.72	24.08	50.73	2.86	50.73
		Control	22.18	3.93	0.38	73.50	4.32	49.62

The test set contains 20 families in which both the target and control questions ask which half of a channel has the higher mean: seven biomedical and thirteen environmental groups. The target

channel is edited and the control channel is unchanged. The remaining 22 groups use a same-channel range control. Table 15 reports the two-channel subset, with equal weight for its two domains. It retains the original tasks, splits, and fitted heads. This within-test-set analysis uses a shared question predicate with unmatched channel difficulty and a different domain composition from the full test set.

Table 15: Two-channel half-mean subset: 20 source groups from biomedical and environmental observations. Each of the two domains receives equal weight, followed by seeds and models. Update and preserve require both answers to be correct.

Readout	Weights	Update	Target flip	Preserve	Control flip
Task-specific	P	14.08	26.34	46.61	11.63
	R	9.91	14.32	59.25	2.31
Question-conditioned	P	0.00	0.00	47.95	0.00
	R	1.85	3.33	55.75	1.98

The native constrained-choice test contains 25 first-to-second target edits and 17 second-to-first edits. Across five models, the original correct option positions are balanced at 105 evaluations each for A and B. Conditioning on initial correctness yields 74 cases at position A and 26 at position B. Table 16 retains these denominators: 70 of the former and 18 of the latter keep the old semantic answer. The total remains 88 of 100. These descriptive counts characterize answer retention under the stated answer-class and option-position distributions.

Table 16: Native constrained-choice counts by target edit direction and original correct option position. Evaluations pool five models on the same 42 families. Correct update and old retained condition on an initially correct target answer.

Gold half	Old position	Evaluations	Initially correct	Update	Old retained
First to Second	A	55	31	2	29
	B	70	9	2	7
Second to First	A	50	43	2	41
	B	35	17	6	11

F Readout, Learning and Edit Controls

F.1 Visible Statistics and Training Budget

The positive control replaces the temporal backbone vector with 20 statistics computed from the visible arrays. It retains the question encoder and shared scoring form. The 100-epoch and 500-epoch fits are distinct training conditions on the same split. Table 17 includes their matched-budget, separately trained question-only controls. This establishes learnability for a statistics input under the specified scorer.

Table 17: Visible-statistics controls for the shared question scorer. Scores are percentages averaged over three seeds. Question-only baselines are trained separately at the corresponding budget.

Model	Input	Epochs	Obs. acc.	All acc.	TUS	CIS	FCJS	QSJA
Qwen3-8B	Stats	100	59.08	56.93	34.20	41.48	1.57	43.20
		500	65.46	63.05	47.24	45.16	10.11	50.86
	Q only	100	47.71	46.27	0.00	45.51	0.00	22.76
		500	48.49	47.16	0.00	44.14	0.00	22.07

Model	Input	Epochs	Obs. acc.	All acc.	TUS	CIS	FCJS	QSJA
Llama-3.1-8B	Stats	100	64.08	62.28	47.08	42.37	8.27	47.92
		500	69.53	68.13	66.54	62.92	39.84	62.19
	Q only	100	49.10	47.75	0.00	43.61	0.00	21.81
		500	49.52	48.38	0.00	43.04	0.00	21.52
Gemma-2-9B	Stats	100	65.27	62.64	44.21	42.29	4.04	47.19
		500	71.37	69.95	60.14	62.11	34.81	60.28
	Q only	100	48.11	47.17	0.00	45.82	0.00	22.91
		500	51.51	49.94	0.00	45.18	0.00	22.59

Table 18: Micro-fit diagnostic on 12 selected validation families, totaling 96 items, evaluated on the fitted items. It measures in-sample fitting capacity.

Model	Seed	Families	Epochs	Initial, %	Fitted, %
Gemma-2-9B	7	12	500	46.88	100.00
Llama-3.1-8B	7	12	500	47.92	100.00
Qwen3-8B	7	12	500	52.08	93.75

Figure 6 shows the test endpoints of the two separate budget conditions. The main learned-readout comparisons retain their 100-epoch budget. The small fitting test uses 12 selected validation families as training data and scores those same items after 500 epochs. Its fitted accuracy tests whether the head can learn the answers on these same items.

F.2 Readout Variants

For Qwen3-8B, Llama-3.1-8B and Gemma-2-9B, the A projection variant inserts a fixed 128-dimensional projection together with its associated normalization. The B variant uses separate task heads. Figure 6 displays the pretrained comparisons on the same domain-macro TUS scale.

Table 19: Readout-variant means, in percent across three seeds. Both weight conditions are retained; full-width A and shared B appear in Table 9.

Model	Variant	Obs. acc.	All acc.	TUS	CIS	FCJS	QSJA
Qwen3-8B	A 128D P	48.79	48.37	0.64	41.83	0.00	23.09
	A 128D R	46.67	48.28	3.13	52.42	0.64	24.43
	B task P	47.63	46.96	0.00	46.25	0.00	23.12
	B task R	47.06	46.04	0.00	52.68	0.00	25.42
Llama-3.1-8B	A 128D P	46.56	49.20	14.08	43.75	4.49	21.78
	A 128D R	45.12	46.84	9.34	39.62	3.21	22.97
	B task P	49.72	49.13	6.70	44.04	3.21	23.30
	B task R	43.72	44.07	6.23	35.35	1.28	20.60
Gemma-2-9B	A 128D P	46.33	48.13	12.59	45.25	4.13	26.26
	A 128D R	47.30	47.13	3.31	43.38	1.19	22.14
	B task P	51.56	52.02	6.89	48.45	3.49	27.08
	B task R	52.72	51.39	0.00	52.39	0.00	26.52

Projection lowers updating for all three task-specific readouts. Separate question-conditioned heads improve Llama and Gemma while leaving Qwen at zero. Projection changes both width and normalization; separate heads supply task identity and increase total parameters. These results show that the measured effect depends on the readout implementation.

F.3 Equal-Gap Edit

Table 20: Original and equal-gap paired scores, in percent, with equal weights for domains. Both edits use the same test groups. Learned-readout entries average three seeds; native entries use deterministic decoding.

Model	Fit	TUS orig.	Gap	CIS orig.	Gap	FCJS orig.	Gap
Qwen3-8B	A P	15.79	13.87	41.76	40.19	7.91	4.13
	A R	10.09	8.26	51.55	51.55	5.96	3.49
	C choice	1.92	1.92	44.69	44.69	1.92	1.92
	C free	3.85	1.92	44.69	41.91	1.92	1.92
	B P	0.00	0.00	45.51	45.51	0.00	0.00
	B R	1.28	1.28	58.19	58.19	0.64	0.64
Llama-3.1-8B	A P	34.77	32.20	35.04	34.40	9.36	8.72
	A R	28.32	24.92	46.85	46.85	19.33	16.22
	C choice	7.42	5.77	43.04	43.83	7.42	3.85
	C free	4.70	3.85	28.21	28.21	4.70	3.85
	B P	0.93	0.93	44.54	44.54	0.93	0.93
	B R	4.04	3.75	38.44	39.63	0.00	0.64
Gemma-2-9B	A P	21.16	12.23	48.25	51.01	10.00	4.68
	A R	3.95	4.59	47.29	46.37	2.38	2.38
	C choice	0.00	0.00	49.60	49.60	0.00	0.00
	C free	0.00	0.00	0.00	0.00	0.00	0.00
	B P	0.64	0.00	46.10	46.10	0.00	0.00
	B R	1.19	3.40	52.30	52.94	1.19	1.83
DeepSeek-V2-Lite	C choice	13.19	9.34	35.56	42.98	1.92	3.85
	C free	0.00	0.00	0.00	0.00	0.00	0.00
OLMo-2-7B	C choice	3.85	1.92	40.78	42.70	3.85	1.92
	C free	3.85	1.92	38.00	42.70	1.92	0.00

The equal-gap edit preserves the absolute half-mean gap on all 42 test groups. It uses the original trained heads or native protocols. The evaluation covers 46 model/interface/seed comparisons spanning selected A/B controls and the five native models. Native TUS is unchanged for Qwen and Gemma and lower for the other three models. Matching the gap magnitude addresses one difficulty dimension while retaining the altered values, offsets and input distributions.

G Reproducibility and Statistical Scope

G.1 Question, Gold, and Output Validation

The evaluation uses a self-contained joint-description question for all 264 evidence families, and shuffled-input gold labels are computed directly from the visible serialized values. Source membership, original arrays, full-evidence golds, option mappings, and data splits are fixed across all reported analyses. All reported outputs correspond to the specified questions, inputs, and gold labels.

The evaluation comprises 210 conditions and 70,560 item-level outputs: 60 task-specific full-evidence fits, 60 question-conditioned full-evidence fits, 60 question-only conditions, and 30 native conditions. Automated validation re-scores the model outputs, verifies all 2,112 visible-value gold labels, and confirms the target-change and control-preservation predicates for all 42 test families. Tables and figures are generated directly from the validated outputs.

G.2 Exploratory Group Statistics

The post-hoc analysis first averages seeds within each family and then resamples source groups separately within each domain. It uses equal domain weights, 10,000 bootstrap draws for percentile intervals, and paired group sign flips for two-sided randomization p values. The bootstrap provides the resampling foundation (Efron, 1979); our domain stratification and group unit follow the sampling structure of this dataset. The finite-sample correction is $(1 + b)/(10{,}000 + 1)$, where b counts draws at least as extreme as the observed result, and the random seed is 20260919. There are 90 tests:

ten models, the three paired metrics, and three contrasts. The contrasts are the pretrained-random effect for each learned interface and the difference between those two effects. Benjamini-Hochberg adjustment covers the complete test family (Benjamini & Hochberg, 1995). These exploratory statistics concern source-group variation for the fixed set of evaluated models.

The statistical results are provided in the supplementary data. Main figures use individual seed effects and descriptive means to display variation across fits. The numerical forecasting tables are based on seed-level summaries.

G.3 Code and Data Availability

The artifact includes complete LaTeX/Bib sources, six vector PDFs, plotting code, numerical inputs, semantic predictions, reconstruction metadata, and a source-linked figure manifest. Training scripts are provided separately from the code that reproduces tables and figures from model outputs. Model weights are excluded from the distributed artifact.

H Additional Related Work

H.1 Forecasting Architectures and Attribution

Forecasting models can acquire useful temporal structure through task-specific design and training. N-BEATS learns residual basis expansions, while N-HiTS uses hierarchical interpolation across scales (Oreshkin et al., 2019; Challu et al., 2022). Linear forecasting baselines also remain competitive in settings where more elaborate attention architectures are often assumed to be necessary (Zeng et al., 2022). These findings make the attribution question concrete: a successful forecasting system may owe its performance to its numerical representation, training target, or readout.

Transformer forecasters encode different assumptions about temporal dependence. Informer makes attention more efficient for long sequences (Zhou et al., 2020). Autoformer and FEDformer incorporate decomposition together with autocorrelation or frequency-domain processing (Wu et al., 2021; Zhou et al., 2022). PatchTST groups observations into patches and shares its model across channels, whereas iTransformer treats variates as tokens (Nie et al., 2022; Liu et al., 2023). TimesNet models within-period and across-period variation through a two-dimensional representation (Wu et al., 2022). Their distinct inputs and architectures explain why comparing complete systems answers a different question from changing the weights within one frozen architecture.

Preprocessing and evaluation also affect the comparison. Reversible instance normalization addresses shifts in the location and scale of a series (Kim et al., 2022). Hewamalage et al. (2022) explain how data partitions, error measures, and aggregation choices shape forecasting conclusions. Our attribution tests keep these choices fixed within each pretrained-random pair and report the numerical head separately from expert selection.

H.2 Representations and Their Readouts

Useful time-series features arise under different learning regimes. TS2Vec learns representations through a hierarchical contrastive objective (Yue et al., 2021). ROCKET combines fixed random convolutional features with a trained classifier (Dempster et al., 2019). Together they motivate a distinction between the source of the features and the supervised map that turns them into answers. A random temporal representation can still support a capable trained readout.

Probe methodology further clarifies this distinction. Minimum-description-length probing considers the cost of learning and transmitting labels from a representation (Voita & Titov, 2020); information-theoretic probing formalizes what a probe can reveal about encoded linguistic structure (Pimentel et al., 2020). Our task-specific heads measure supervised accessibility under the stated protocol. Comparing those heads with a question-conditioned scorer and native decoding then asks how that accessibility translates into different uses of the same backbone.

H.3 Behavior Under Controlled Changes

Behavioral evaluation tests the relation between predictions as well as their individual correctness. Semantically equivalent adversarial rules target changes that should preserve an answer (Ribeiro et al., 2018). Contrast sets instead make small, meaningful changes near a model's decision boundary (Gardner et al., 2020). Counterfactually augmented data use such edits during training (Kaushik et al., 2019); here the edits evaluate existing models. Our evidence families combine a fact that must change with a control fact that must remain correct, using one visible numerical edit.

Heuristic evaluations show that correct predictions can coexist with reliance on an unintended shortcut (McCoy et al., 2019). Underspecification studies likewise show that similar predictive scores can conceal different behavior under later tests (D'Amour et al., 2020). These observations motivate reporting updating, preservation, and joint success alongside item accuracy. The paired outcomes identify which evidence relations the answers satisfy.