TREFIN: INTERPRETABLE FINANCIAL FORECASTING THROUGH TYPED STATE DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

An interpretable financial forecast should expose the measurable state on which it depends. We introduce TREFIN (Typed Relational Evolution for Financial Forecasting), a probabilistic framework that links asset concepts, residual dependence, and market conditions through an observed-factor measurement procedure. Concepts and a signed residual graph evolve jointly, with distinct propagation rules for directional and risk-like quantities. Future measurements supervise the predicted states, which provide the sole information interface to an asymmetric, heavy-tailed return decoder. A system-level theorem establishes invariant state domains, distributional validity, finite-horizon error propagation, and a conditional downside-scale response. On the 32-asset FinMultiTime task, the validation-calibrated model achieves the lowest mean squared and root mean squared errors and ranks among the top two methods on all six metrics in the main five-seed comparison. State diagnostics reveal how named quantities affect the forecast, while chronological evaluations and retraining expose complementary strengths and remaining calibration limitations. TREFIN makes intermediate financial states testable without giving up competitive nonlinear prediction.

1 INTRODUCTION

Two financial forecasts can predict similar expected returns and imply very different risks. One may reflect persistent price movement in a liquid market; another may arise during elevated downside variation and unusually strong dependence among assets. A point prediction alone does not distinguish these conditions. As Cont (2001) documents, heavy tails, clustered volatility, and heterogeneous dependence are central features of financial returns. Weak directional predictability can coexist with persistent variation in risk. The efficient-market perspective of Fama (1970) likewise makes predictability conditional on available information and the model of expected returns. A useful forecasting representation should therefore distinguish return direction, risk magnitude, and dependence, rather than compress them into an undifferentiated hidden vector.

Expressive learning is valuable for this problem. Gu et al. (2020) demonstrate gains from nonlinear predictor interactions, and Kelly et al. (2024) show that appropriately regularized complexity can improve return prediction. Economic structure need not oppose flexibility: Chen et al. (2024) combine neural learning with asset-pricing restrictions. Recent forecasting models improve temporal and cross-variable representations, including patching in Nie et al. (2023), variate-centered attention in Liu et al. (2024), and market-guided stock modeling in Li et al. (2024). The remaining question is not whether context or dependence matters. It is whether the state that produces a forecast has operational financial meanings, whether its predicted evolution can be checked, and whether its designated risk effects follow those meanings.

Concept bottleneck models introduced by Koh et al. (2020) offer a natural starting point by routing predictions through human-understandable variables. However, financial concepts are continuous, dependent estimates rather than error-free semantic labels. The leakage analysis of Havasi et al. (2022) further shows that soft concepts may transmit information beyond their intended meanings. Calling a coordinate volatility is not sufficient: its measurement, transformations, and role in the output must be specified. A future concept trajectory should also be evaluated against subsequent measurements, not justified solely by accurate return predictions.

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.

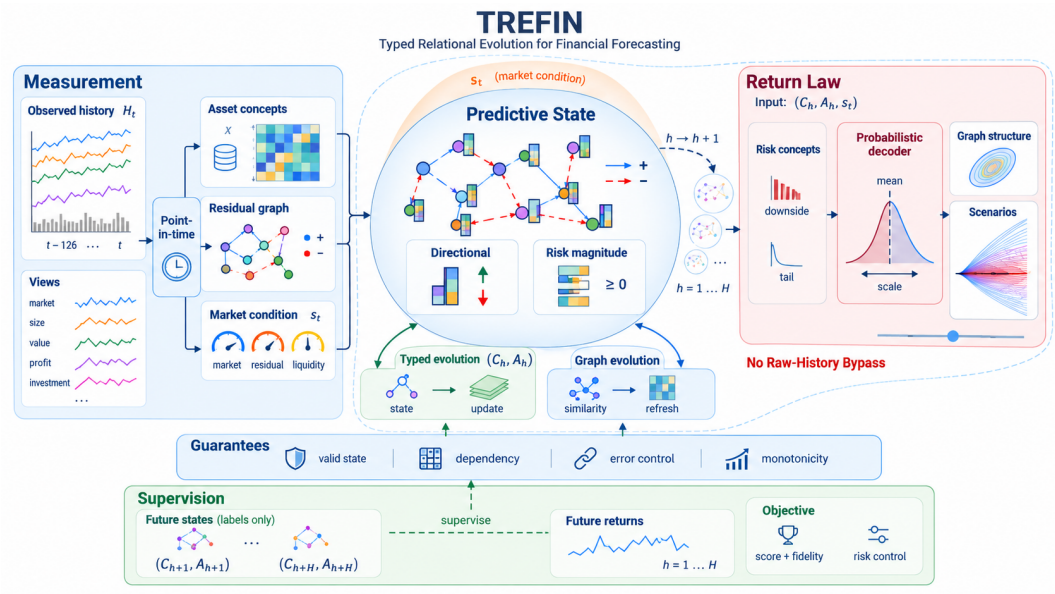


Figure 1: TREFIN overview. At origin t, available history $\mathcal{H}_t$ yields an N-asset, K-concept state $\mathbf{C}_0$, signed residual graph $\mathbf{A}_0$, and market vector $\mathbf{s}_t$. Evolved states $(\mathbf{C}_h, \mathbf{A}_h)$ at lead $h \leq H$ feed the return law; H is the prediction horizon and hats are suppressed in the schematic. Future measurements enter training losses only. Non-neural measurement remains subject to train-fitted preprocessing. Signed edges denote associations, and risk typing concerns propagation, not positivity of standardized coordinates. Risk masking constrains designated scale/dependence channels; the mean can use the full state.

A financially typed predictive state. Our hypothesis is that *measurable financial quantities can retain useful short-horizon predictive information while supporting learned state evolution and distributional responses that remain testable*. Here a state is the model's interface to available history; a type specifies a coordinate's interpretation and admissible treatment. Negative dependence may reverse a directional signal, but should not turn a volatility magnitude into negative risk. This distinction motivates different relational operators for directional and risk-like coordinates. It restricts the information route while preserving nonlinear learning within it, without assuming that the chosen state contains all predictive information.

TREFIN implements this principle through a common observed-factor construction. Estimated factor exposures enter asset concepts, factor residuals define a signed dependence graph, and aggregate residual dependence helps characterize market conditions. The concept and graph states then evolve together. Future measurements supervise their trajectory, and an asymmetric return decoder uses only this interface. Selected risk coordinates enter downside scale through an explicitly positive channel. Consequently, the quantities used for prediction are also the quantities whose evolution and designated output effects can be inspected. Interpretation is attached to the computational mechanism, following the architectural perspective of Rudin (2019), rather than inferred from a separate explanation model.

Figure 1 summarizes this shared measurement–evolution–prediction route and its training supervision.

The main FinMultiTime comparison supports this restricted interface with state-of-the-art squared error among the evaluated methods and a top-two ranking on all six reported metrics. The system-level analysis establishes a valid finite-horizon mechanism, while the experiments examine predictive quality, state fidelity, and component contributions separately. Our contributions are the financially grounded state construction, its type-aware evolution and constrained distributional readout, and their joint theoretical and empirical evaluation. The resulting model connects measurement, prediction, and explanation through a shared state, instead of assigning independent roles to an assortment of interpretable components.

2 RELATED WORK

Financial measurements provide anchors for interpretation: Fama & French (2015) describe common-factor exposures, Amihud (2002) measures illiquidity, and Barndorff-Nielsen et al. (2010) distinguishes downside variation. The factor-plus-residual perspective of Fan et al. (2013) motivates separating systematic exposure from remaining dependence. These measurements do not themselves specify a learned forecast of their future evolution. Conversely, forecasting advances such as TimeXer by Wang et al. (2024b), TimeMixer++ by Wang et al. (2025a), and Timer-XL by Liu et al. (2025a) do not require their internal coordinates to follow registered financial measurements. Concept bottlenecks and embeddings, studied by Koh et al. (2020) and Espinosa Zarlenga et al. (2022), make semantics part of prediction, while Rasul et al. (2021) and Solano Carrillo et al. (2026) develop expressive probabilistic forecasting. TREFIN connects these concerns through a measured, supervised state trajectory with type-specific propagation and constrained risk responses. Appendix A develops the full comparison across financial modeling, forecasting, concept-based interpretation, and residual graphs.

3 METHODOLOGY

3.1 FORECASTING THROUGH MEASURED FINANCIAL STATES

Let N be the number of assets, K the number of concepts per asset, and H the forecast horizon. The forecast origin t is the last observation available for prediction. Asset indices are $i, j \in \{1, \dots, N\}$, concept indices are $k \in \{1, \dots, K\}$, and return leads are $h \in \{1, \dots, H\}$. With positive recorded price $P_{t,i}$ and observed one-period log risk-free return r_t^{rf}, the excess log return and supervised target matrix are

$$r_{t,i} = \log(P_{t,i}/P_{t-1,i}) - r_t^{\mathrm{rf}}, \qquad [\mathbf{Y}_t]_{h,i} = r_{t+h,i}, \quad \mathbf{Y}_t \in \mathbb{R}^{H\times N}. \tag{1}$$

These are H future one-period returns, not a cumulative H-period return. All returns are in decimal units. The input consists of measured concepts $\mathbf{C}_{t,0} \in \mathbb{R}^{N\times K}$, a symmetric signed residual graph $\mathbf{A}_{t,0} \in \mathbb{R}^{N\times N}$ with zero diagonal, and market conditions $\mathbf{s}_t \in \mathbb{R}^3$. Reapplying the same causal measurements at time $t + h$ defines future concept and graph targets $\mathbf{C}_{t,h}$ and $\mathbf{A}_{t,h}$. Those targets supervise training but never enter a forecast as inputs.

One decomposition, three complementary descriptions. At each origin, an exponentially weighted ridge regression of each asset's past excess returns on observed factors yields estimated exposures and factor residuals. The exposures are coefficients estimated from data, not neural parameters. Residuals are returns after subtracting the fitted factor contribution and intercept. A regularized precision estimate of standardized residuals gives partial correlations; signed thresholding yields the initial graph. Thus the graph represents dependence remaining under the specified factor adjustment, not raw return correlation or identified causal transmission.

The principal model has $K = 26$ concepts: momentum, upside/downside log-semivariance, jump share, and log illiquidity at lookbacks $L \in \{5, 21, 63, 126\}$, plus signed-volume flow and five factor exposures. Here L denotes a lookback length; $w_u^{(L)} \propto \exp(-3u/L)$ denotes its normalized weight at historical lag $u \in \{0, \dots, L-1\}$.

For a regression lookback L_g, let $\mathbf{f}_q \in \mathbb{R}^{d_f}$ be the d_f observed factor returns at date q, and let $\lambda_\beta > 0$ be the fixed ridge penalty. The fitted intercept $\widehat{\alpha}_{t,i}$ and exposure vector $\widehat{\boldsymbol{\beta}}_{t,i}$ minimize

$$\sum_{u=0}^{L_g-1} w_u^{(L_g)}(r_{t-u,i} - \alpha - \boldsymbol{\beta}^\top \mathbf{f}_{t-u})^2 + \lambda_\beta \|\boldsymbol{\beta}\|_2^2 \quad \text{over } \alpha \in \mathbb{R},\ \boldsymbol{\beta} \in \mathbb{R}^{d_f}, \tag{2}$$

where $\|\cdot\|_2$ is Euclidean norm. For $q \in \{t - L_g + 1, \dots, t\}$, this yields residuals $\varepsilon_{q,i}^{(t)} = r_{q,i} - \widehat{\alpha}_{t,i} - \widehat{\boldsymbol{\beta}}_{t,i}^\top \mathbf{f}_q$. Let $\widehat{\boldsymbol{\Theta}}_t$ denote the positive-definite Graphical Lasso precision estimate of standardized residuals, following Friedman et al. (2008). For $i \neq j$, residual partial correlation is

$$\rho_{t,ij}^{\varepsilon} = -\widehat{\Theta}_{t,ij}/\sqrt{\widehat{\Theta}_{t,ii}\widehat{\Theta}_{t,jj}}. \tag{3}$$

The diagonal entries in the denominator normalize the conditional association. Smooth signed thresholding of this matrix supplies the measured graph. Market conditions are market log realized variance, mean absolute residual partial correlation, and market-weighted log illiquidity. The shared decomposition separates common-factor exposure, remaining cross-asset structure, and the aggregate environment. Jump and flow coordinates remain operational daily proxies. Appendix B.2 gives every estimator, normalization, safeguard, and availability rule.

Training-only robust scaling defines coordinate bounds $b_k^- \le C_{ik} \le b_k^+$ and the concept box $\mathcal{C}$. The graph set $\mathcal{A}$ contains symmetric zero-diagonal matrices whose Frobenius norm is at most the train-estimated radius $B_A > 0$; the Frobenius norm is the square root of the sum of squared entries. The operators $\Pi_{\mathcal{C}}$ and $\Pi_{\mathcal{A}}$ are Euclidean projections onto these sets. We initialize predicted states with the measurements, $\widehat{\mathbf{C}}_{t,0} = \mathbf{C}_{t,0}$ and $\widehat{\mathbf{A}}_{t,0} = \mathbf{A}_{t,0}$, and hold $\mathbf{s}_t$ fixed over the short horizon. Hats distinguish predictions from measured targets. Below, we suppress only the fixed origin index t in predicted states.

3.2 Types determine the joint state dynamics

For any admissible graph $\mathbf{A}$, define the signed and unsigned Laplacians

$$\mathbf{L}_s(\mathbf{A}) = \operatorname{diag}(|\mathbf{A}|\mathbf{1}_N) - \mathbf{A}, \qquad \mathbf{L}_u(\mathbf{A}) = \operatorname{diag}(|\mathbf{A}|\mathbf{1}_N) - |\mathbf{A}|. \tag{4}$$

Here $|\mathbf{A}|$ is entrywise absolute value, $\mathbf{1}_N$ is the all-one vector, and diag makes a diagonal matrix from a vector. Let $d_k = 1$ for directional concepts (momentum, flow, and exposures) and $d_k = 0$ for risk-like concepts (both semivariances, jump share, and illiquidity). For column $\mathbf{C}_{:,k}$, the typed operator is

$$(\mathcal{L}_{\mathbf{A}}[\mathbf{C}])_{:,k} = \{d_k \mathbf{L}_s(\mathbf{A}) + (1 - d_k)\mathbf{L}_u(\mathbf{A})\}\mathbf{C}_{:,k}. \tag{5}$$

Signed coupling regards opposing directional values across a negative edge as agreement; magnitude-based coupling smooths risk-like coordinates without reversing their financial meaning. Both Laplacians are positive semidefinite, meaning their quadratic forms are nonnegative (Lemma 1). This property motivates the propagation rule independently of predictive training.

Let $\boldsymbol{\gamma}(\mathbf{s}_t) \in \mathbb{R}_+^K$, $D(\mathbf{s}_t) > 0$, and $\mathbf{b}(\mathbf{s}_t) \in \mathbb{R}^K$ be bounded neural functions specifying damping, diffusion strength, and common forcing. The positive Euler step Δ and graph refresh fraction $\eta_A \in (0, 1]$ are fixed hyperparameters. A bounded nonlinear correction $\mathbf{F}_h \in \mathbb{R}^{N \times K}$ uses each asset's current concepts, graph-aggregated concepts, market conditions, and learned lead embedding $\mathbf{e}_h$. A symmetric zero-diagonal proposal $\mathbf{G}_h$ uses the current concepts, market conditions, and the same embedding. For transition steps $h \in \{0, \dots, H-1\}$,

$$\widehat{\mathbf{C}}_{h+1} = \Pi_{\mathcal{C}}\Big[\widehat{\mathbf{C}}_h + \Delta\Big\{-\widehat{\mathbf{C}}_h \operatorname{diag}(\boldsymbol{\gamma}(\mathbf{s}_t)) - D(\mathbf{s}_t)\mathcal{L}_{\widehat{\mathbf{A}}_h}[\widehat{\mathbf{C}}_h] + \mathbf{1}_N \mathbf{b}(\mathbf{s}_t)^\top + \mathbf{F}_h\Big\}\Big], \tag{6}$$

$$\widehat{\mathbf{A}}_{h+1} = \Pi_{\mathcal{A}}\big[(1 - \eta_A)\widehat{\mathbf{A}}_h + \eta_A \mathbf{G}_h\big]. \tag{7}$$

Both updates read the same old state; the graph proposal never reads $\widehat{\mathbf{C}}_{h+1}$. Specifically, separate neural encoders produce vectors $\mathbf{u}_{h,i}, \mathbf{v}_{h,i} \in \mathbb{R}^{d_g}$ from an asset's current concepts, the market state, and lead embedding, where d_g is the graph-embedding dimension. They define affinity $S_{h,ij}$ and proposal entries

$$S_{h,ij} = \frac{\mathbf{u}_{h,i}^\top \mathbf{u}_{h,j} - \mathbf{v}_{h,i}^\top \mathbf{v}_{h,j}}{\sqrt{d_g}}, \quad G_{h,ij} = \tanh(S_{h,ij})\sigma((|S_{h,ij}| - c_G)/\tau_G), \quad i \neq j, \tag{8}$$

with $G_{h,ii} = 0$, fixed gate center $c_G > 0$, temperature $\tau_G > 0$, and sigmoid $\sigma(x) = (1 + e^{-x})^{-1}$. The difference of Gram matrices represents positive and negative affinities while preserving symmetry. All nodewise networks share weights across assets. Damping controls persistence, typed diffusion transfers relational information, and bounded forcing supplies nonlinear flexibility within the projected domain. Figure 1 distinguishes this predictive route from its training-only supervision. Appendix B.3 specifies the networks and their parameter bounds.

3.3 A distribution whose direct risk response is inspectable

The state at lead h is decoded with embedding $\mathbf{e}_h$; its preceding transition used $\mathbf{e}_{h-1}$. The decoder predicts a bounded mean parameter $\mu_{h,i}$, positive upside/downside scales $\sigma_{h,i}^+, \sigma_{h,i}^-$, and Student

degrees of freedom $\nu_h > 2$. Let $\mathcal{R}$ contain the downside-semivariance, jump-share, and illiquidity coordinates. To isolate their direct effect, a feature vector $\boldsymbol{\psi}_{h,i}^{\text{free}}$ masks these coordinates in both the local concepts and their graph aggregate, while retaining the market state and lead. With train-fitted return scale $a_i^r > 0$, fixed scale floor $a_0 > 0$, learned scalar network f_-, and learned positive weights $w_k^{\mathcal{R}}$,

$$\sigma_{h,i}^- = a_i^r \left[a_0 + \text{softplus}\left(f_-(\boldsymbol{\psi}_{h,i}^{\text{free}}) + \sum_{k \in \mathcal{R}} w_k^{\mathcal{R}} \widehat{C}_{h,ik} \right) \right], \tag{9}$$

where $\text{softplus}(x) = \log(1 + e^x)$. The weights are a positive softmax allocation with bounded total strength. Holding the graph, other concepts, market state, and lead fixed therefore gives $\partial \sigma_{h,i}^- / \partial \widehat{C}_{h,ik} > 0$ for every $k \in \mathcal{R}$. The guarantee concerns this direct decoder response; the mean can use the full state. Masking both local and aggregated risk features is essential, since leaving the latter unmasked would provide an unconstrained route around the positive channel.

The graph also determines cross-sectional dependence. With learned baseline precision vector $\mathbf{q}_h$ having entries above the fixed floor $\epsilon_Q > 0$, learned coupling $\lambda_{Q,h} > 0$, and identity matrix $\mathbf{I}_N$, define

$$\mathbf{Q}_h = \text{diag}(\mathbf{q}_h) + \lambda_{Q,h} \mathbf{L}_s(\widehat{\mathbf{A}}_h) + \epsilon_Q \mathbf{I}_N. \tag{10}$$

Positive semidefiniteness of the signed Laplacian makes $\mathbf{Q}_h$ strictly positive definite. Normalizing $\mathbf{Q}_h^{-1}$ supplies the dependence matrix for a symmetric Student innovation $\boldsymbol{\xi}_h \in \mathbb{R}^N$. The precision and tail heads use risk-masked features. In the ideal unit-marginal-variance law, define $[x]_+ = \max(x, 0)$ and $\kappa(\nu_h) = \mathbb{E}[[\xi_{h,i}]_+]$, where $\mathbb{E}$ denotes expectation. A generated excess-return scenario is

$$\widehat{y}_{t,h,i} = \mu_{h,i} + \sigma_{h,i}^+ [\xi_{h,i}]_+ - \sigma_{h,i}^- [-\xi_{h,i}]_+ - \kappa(\nu_h)(\sigma_{h,i}^+ - \sigma_{h,i}^-). \tag{11}$$

The correction separates mean from asymmetric shock scales. A shared Student mixing variable couples assets within each lead, while innovations at different leads are conditionally independent. Appendix B.4 gives the complete sampler, including the registered numerical jitter and its small mean-correction boundary. No raw-history path bypasses the concept–graph–market interface.

3.4 Joint supervision and a system-level guarantee

For an observed cross-sectional return vector $\mathbf{y} \in \mathbb{R}^N$ and $S \geq 2$ independent model scenarios $\widehat{\mathbf{y}}^{(s)}$, indexed by $s \in \{1, \ldots, S\}$, the empirical energy score is

$$\mathcal{E}_S = \frac{1}{S} \sum_{s=1}^{S} \|\widehat{\mathbf{y}}^{(s)} - \mathbf{y}\|_2 - \frac{1}{2S(S-1)} \sum_{s \neq s'} \|\widehat{\mathbf{y}}^{(s)} - \widehat{\mathbf{y}}^{(s')}\|_2, \tag{12}$$

where the second sum includes all ordered unequal pairs. Following the proper-scoring framework of Gneiting & Raftery (2007), we average this cross-sectional score over training origins and leads to obtain $\mathcal{L}_{\text{ES}}$. Let $\mathcal{L}_C$ and $\mathcal{L}_A$ be normalized Huber losses between predicted and measured concept and graph paths, respectively; a Huber loss is quadratic for small errors and linear for large errors. Let $\mathcal{L}_{\text{gain}}$ be the soft state-sensitivity penalty. With nonnegative fixed weights $\lambda_C, \lambda_A, \lambda_G$ and neural parameter vector θ, the effective training objective is

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{ES}} + \lambda_C \mathcal{L}_C + \lambda_A \mathcal{L}_A + \lambda_G \mathcal{L}_{\text{gain}}. \tag{13}$$

The implementation also records a monotonicity penalty that is identically zero under the positive risk parameterization. Auxiliary losses include anchored lead zero and all future leads, with train-estimated increment scales balancing coordinates. The targets, rolling factor coefficients, measured graphs, and scalers are fixed during neural optimization. Future-state supervision penalizes deviations from named quantities, while the proper score trains the return distribution. The two tests are complementary: accurate returns do not guarantee faithful states, and faithful states do not guarantee accurate returns. Appendix B.5 specifies every loss reduction and coefficient.

From admissible states to admissible forecasts. For fixed trained parameters and market conditions, compare the rollout with any admissible reference path $(\mathbf{C}_h^\star, \mathbf{A}_h^\star)$. Define its two-block error vector $\boldsymbol{\delta}_h = (\|\widehat{\mathbf{C}}_h - \mathbf{C}_h^\star\|_F, \|\widehat{\mathbf{A}}_h - \mathbf{A}_h^\star\|_F)^\top$. Let $\mathbf{M} \in \mathbb{R}_+^{2\times 2}$ upper-bound the four transition-block sensitivities uniformly over the state domain and leads. Let $\boldsymbol{\varepsilon}_q \in \mathbb{R}_+^2$ bound the concept and graph discrepancies between reference state $q+1$ and one learned transition from reference state q. Finite such sensitivities exist for the fixed finite-weight networks on the compact domains.

Theorem 1 (Finite-horizon state-to-distribution contract) *Starting from an admissible measured state, every TREFIN rollout remains in $\mathcal{C} \times \mathcal{A}$ and produces a valid cross-sectional return law with positive scales and finite first moment. For each $h \in \{1, \dots, H\}$, the componentwise error satisfies*

$$\boldsymbol{\delta}_h \leq \mathbf{M}^h \boldsymbol{\delta}_0 + \sum_{q=0}^{h-1} \mathbf{M}^{h-1-q} \boldsymbol{\varepsilon}_q. \tag{14}$$

Here q indexes transition steps and $\mathbf{M}^0$ is the 2×2 identity. At fixed other decoder inputs, each designated risk coordinate increases its asset's downside scale.

Proof sketch. Euclidean projection gives state invariance; Laplacian positivity and parameter floors give valid probability parameters. Iterating $\boldsymbol{\delta}_{q+1} \leq \mathbf{M}\boldsymbol{\delta}_q + \boldsymbol{\varepsilon}_q$ yields the bound. Positive risk weights and the positive derivative of softplus give the direct response. Appendix C proves the expanded result, with explicit constants and decoder regularity conditions for a distribution-error extension. The bound separates initial measurement error from transition mismatch; it does not require either error to contract. The training penalty encourages lower sensitivity but is not a global certificate.

The same decomposition connects state fidelity to distributional error under an explicit decoder condition. Let W_1 be the minimum expected Euclidean distance over couplings of two probability laws. Suppose the fixed decoder is L_D-Lipschitz in W_1 under the sum of the concept and graph norms, where $L_D \geq 0$ is uniform over leads. Let $\widehat{\mathbb{P}}_h$ and $\mathbb{P}_h^\star$ be the forecast and reference laws, and let $\varepsilon_D \geq 0$ bound the decoder's error at the reference states. With $\mathbf{1}_2 = (1, 1)^\top$, the triangle inequality gives

$$W_1(\widehat{\mathbb{P}}_h, \mathbb{P}_h^\star) \leq L_D \mathbf{1}_2^\top \boldsymbol{\delta}_h + \varepsilon_D. \tag{15}$$

Together with equation (14), this separates initial measurement error, accumulated transition mismatch, and decoder approximation. State-path supervision addresses a different source of forecast error from return scoring; neither objective substitutes for the other.

Selection and prediction. Checkpoint selection uses validation energy score. An optional scalar $\varrho \in [0.05, 1]$ then shrinks only the structural mean: $\widehat{\mathbf{y}}^{(s)}(\varrho) = \varrho \boldsymbol{\mu}^{\mathrm{str}} + \boldsymbol{\epsilon}^{(s)}$, where $\boldsymbol{\mu}^{\mathrm{str}}$ is the pre-calibration mean vector and $\boldsymbol{\epsilon}^{(s)}$ its scenario innovation. Pairwise scenario differences are unchanged, so the validation objective is convex in ϱ. After selection, a fresh model reloads the checkpoint before the final test evaluation. All measurement fitting and tuning exclude test outcomes (Algorithm 1).

4 Experiments

4.1 Protocol and predictive performance

We use the numerical U.S. panel of FinMultiTime introduced by Xu et al. (2025), with five observed Fama–French factors, $N = 32$ assets, and $H = 5$ daily excess-return leads. The main split trains on 2009–2017, validates on 2018–2019, and tests on 2020–2021, with 500 common test origins. Training seeds are $\{1, 7, 19, 37, 73\}$. TREFIN has 95,258 trainable parameters and hidden width 64. AdamW uses learning rate 10^{-3}, weight decay 10^{-4}, batch size 64, a 20-epoch ceiling, and validation-based patience six. Training, validation, and testing use 16, 64, and 256 scenarios, respectively. The transition uses $\Delta = 0.20$ and $\eta_A = 0.10$; loss weights are $\lambda_C = 0.25$, $\lambda_A = 0.08$, and $\lambda_G = 0.10$. A cosine learning-rate schedule and gradient-norm clipping at one control optimization. Appendix E.2 specifies the purged boundaries, source provenance, and complete configuration.

We compare statistical and neural controls with recent forecasting methods. The task-adapted implementations follow official model settings; Timer-XL is trained on this task. Probabilistic adapters generate the scenarios used for energy scoring. This does not imply that every original backbone has a native probabilistic head. We report mean squared error (MSE), its square root (RMSE), mean absolute error (MAE), energy score (ES), and the released multi-step information coefficient and ratio (msIC/msIR). The last two use Pearson correlation along the five-lead horizon, not cross-sectional rank correlation. Appendix E.1 defines all reductions and the comparison's provenance boundaries.

Leading squared error with an interpretable information route. Table 1 shows that calibrated TREFIN achieves the best MSE and RMSE among the evaluated methods and is second on each

Table 1: FinMultiTime U.S., fixed main split: five-seed means. Bold and underline mark the best and second-best means, not significance. Full seed deviations appear in Tables 4 and 5.

Method Multiplier	MSE ↓ 10^4	RMSE ↓ 10^2	MAE ↓ 10^2	ES ↓ 10^2	msIC ↑ 10^2	msIR ↑ 10^2
ShrunkVAR	7.2052	2.6842	1.7456	9.4520	0.3124	0.5510
HistGradientBoosting	7.1824	2.6800	1.7325	9.5015	0.3541	0.6234
StateMLP	7.1513	2.6742	1.7258	9.3842	0.3805	0.6845
RawHistoryMLP	7.2504	2.6927	1.7602	9.6055	0.2501	0.4102
GRU	7.0856	2.6619	1.7104	9.2560	0.4208	0.7850
DLinear	6.9852	2.6430	1.6985	9.1540	0.4855	0.9120
PatchTST	6.9524	2.6367	1.6912	9.1235	0.5120	0.9854
iTransformer	6.9215	2.6309	1.6854	9.0842	0.5405	1.0540
MASTER	6.9408	2.6345	1.6885	9.1055	**0.6125**	1.1450
TimeXer	6.8954	2.6259	1.6815	9.0520	0.5558	1.0850
TimeMixer++	6.8625	2.6196	1.6758	9.0185	0.5704	**1.1985**
Timer-XL	6.8552	2.6182	1.6724	8.9950	0.5755	1.1520
TEDM	<u>6.8415</u>	<u>2.6156</u>	**1.6685**	**8.9650**	0.5650	1.1305
TREFIN (uncal.)	6.8542	2.6180	1.6764	9.0060	<u>0.5864</u>	<u>1.1915</u>
TREFIN (cal.)	**6.8171**	**2.6110**	<u>1.6704</u>	<u>8.9744</u>	<u>0.5864</u>	<u>1.1915</u>

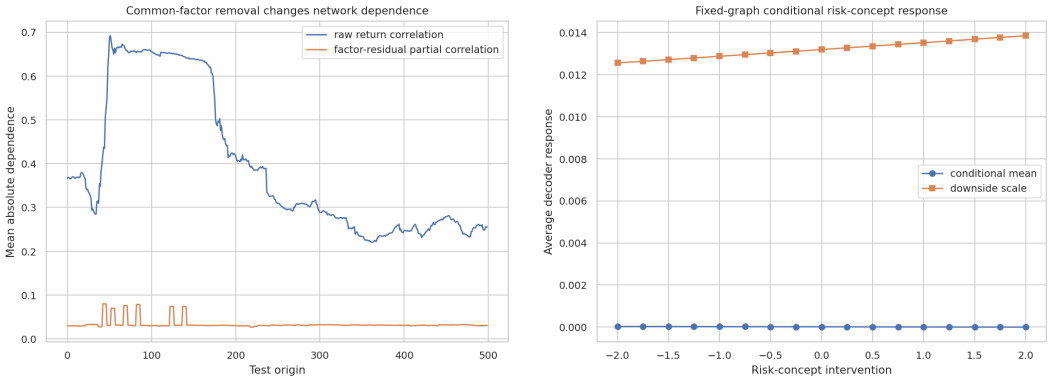


Figure 2: Inspectable state and response, fold 1, seed 1. Left: raw return correlation versus factor-residual partial dependence. Right: mean and downside-scale responses to a standardized downside-semivariance coordinate at the first test origin, with all other decoder inputs fixed.

remaining metric. Relative to TEDM, the strongest squared-error comparator, the reductions are 0.36% in MSE and 0.18% in RMSE. The important combination is this consistently competitive profile with a measured state bottleneck and constrained direct risk responses, rather than dominance on every metric. Full five-seed deviations appear in Tables 4 and 5; rankings alone are not significance tests.

Validation-only shrinkage reduces displayed MSE from 6.8542 to 6.8171 and ES from 9.0060 to 8.9744, while preserving msIC/msIR. It also reduces MSE's across-seed standard deviation from 0.0414 to 0.0066. This is consistent with correcting the amplitude of a weak mean signal, not changing its ordering. Three expanding-window folds comprise 15 fitted models: shrinkage helps MSE and ES in the first two periods but slightly worsens them in the third (Table 6). The main-table advantage is therefore a fixed-task result, with temporal variation evaluated separately.

4.2 Interpretability and the Roles of State Components

Figure 2 shows two directly inspectable aspects of the mechanism. Raw return correlation and residual partial dependence follow substantially different trajectories, distinguishing shared market variation

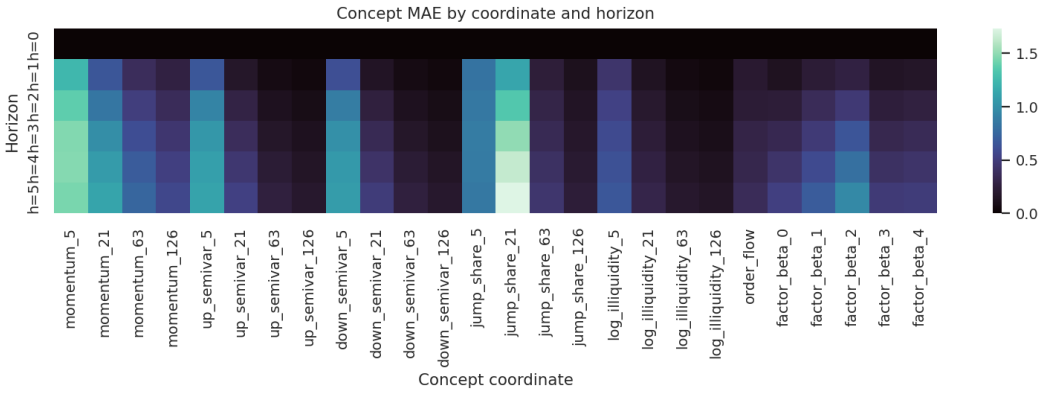


Figure 3: Concept-path fidelity, fold 1, seed 1. Absolute concept errors are averaged over test origins and assets in robust-standardized units. Lead zero is anchored input agreement; subsequent rows evaluate forecasts of the same measured quantities.

Table 2: State interventions and structural retraining. Frozen-model rows are five-seed calibrated means; retrained rows are an uncalibrated seed-1 pair. The two blocks answer different questions. Full statistics are in Tables 7 and 10.

Setting	MSE $\times 10^4$	MAE $\times 10^2$	ES $\times 10^2$	msIC $\times 10^2$
Frozen-model input interventions: five seeds				
Full calibrated	6.8171	1.6704	8.9744	0.5864
Initial graph zero	6.8204	1.6711	9.0795	0.2142
Market at median	6.8208	1.6711	8.9761	0.3310
Risk at median	6.8180	1.6705	9.0305	0.4433
Structural retraining: seed 1, uncalibrated				
Full reference	6.9169	1.6868	9.0526	0.7330
Graph-free	6.8495	1.6750	9.0557	0.9333

from dependence remaining under the factor-and-precision pipeline. The decoder intervention changes one standardized downside-semivariance coordinate while fixing the remaining inputs. Downside scale responds positively as designed, whereas the mean has an unrestricted state-dependent response. These are properties of the fitted predictor, not causal effects of changing market risk. Coordinate-wise forecast errors and graph-support overlap quantify the fidelity of the predicted state trajectories in Appendix E.5.

Figure 3 tests the named concept trajectories against subsequent measurements. The anchored input row is exact by construction; future rows reveal heterogeneous approximation error. The 21-observation jump-share proxy is among the most difficult coordinates, while several long-window measurements change more slowly. This distinction matters for interpretation: a plausible concept name does not imply accurate prediction of that concept. The corresponding graph-support overlap declines to approximately 0.59 at lead five (Appendix E.5). These diagnostics expose where the measured-state approximation is strongest and where it requires improvement, without treating target graphs as observed causal networks.

Dependence of a fitted model versus necessity after retraining. Table 2 separates these two questions. Removing the initial graph from frozen calibrated models increases ES by about 1.17% and reduces msIC; replacing risk coordinates increases ES by 0.63%. The fitted distribution thus uses both interpretable channels. Standardized zero denotes the training median, not zero economic risk, and the graph may regenerate during rollout. A genuinely graph-free retraining pilot instead forces the graph to zero throughout the model. It shows partial degradation in ES alongside improved point errors. Its paired ES difference has a 95% block-bootstrap interval spanning zero, so the single-seed result is a useful trade-off, not a reliable across-seed graph advantage (Appendix E.3).

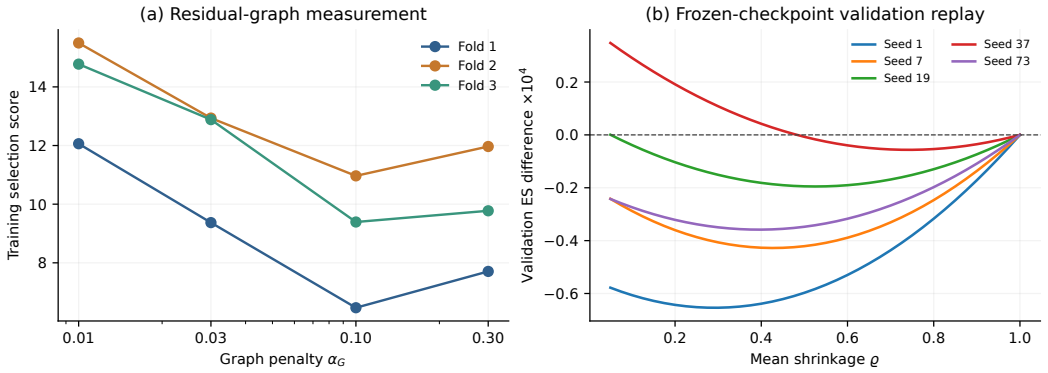


Figure 4: Data-selected controls. Left: training-only residual-graph selection across three folds. Right: validation energy-score differences relative to no mean shrinkage, across five frozen checkpoints and 39 scalar settings. Each curve uses fixed scenarios; no test outcomes select either control.

Testing the structural predictions. Across 15 first-test-origin rollout audits, graph symmetry and diagonal errors and concept-box violations are zero; the minimum precision eigenvalue is at least 0.1048. Fixed-graph risk interventions have minimum positive finite-difference slope at least 3.01×10^{-4}. These checks agree with Theorem 1. Local four-step power-iteration sensitivity estimates are below one at all 1,800 sampled states (maximum 0.9740), but are not upper bounds. The theoretical contribution is finite-horizon validity and error accounting, not a claim of global contraction from these samples (Appendix E.6).

4.3 Sensitivity, efficiency, and scope

Figure 4 separates two data-selected controls. The graph-measurement penalty α_G determines sparsity and is selected on training origins; 0.10 is chosen in each fold from $\{0.01, 0.03, 0.10, 0.30\}$. The validation-only scalar ϱ adjusts mean amplitude after training. Replaying five frozen checkpoints at 39 scalar settings yields convex score-difference curves, as predicted by the common-translation construction. Selected five-seed mean scalars vary across folds (0.478, 0.430, 0.072), reflecting period-dependent signal strength. These measurements assess the implemented selection mechanisms; architecture and loss-weight sweeps remain separate experiments.

Training on the archived RTX 4090 averages 74.15 ± 10.69 seconds across 15 uncalibrated fits, including epoch evaluation but excluding state preprocessing; calibrated runs take 77.98 ± 11.24 seconds. The small neural parameter count does not remove the dense decoder's cubic asset-dimension factorization cost. An hourly BTCF spot/futures refit obtains RMSE 0.005527 and ES 0.003833 over 8,597 test origins. This establishes applicability after adapting measurements and refitting, not zero-shot transfer (Appendices E.7 and E.8).

The nominal 90% intervals cover 80.12%, 85.79%, and 86.54% of held-out scalar returns in the three U.S. periods. Mean shrinkage preserves scenario dispersion, so good squared error and ES do not imply fully calibrated tails. Improving distributional shape calibration and completing multi-seed structural ablations are natural next steps. The present evidence supports competitive predictive accuracy with measurable intermediate forecasts and verifiable direct risk responses.

5 Conclusion

TREFIN connects prediction and interpretation through a financially typed state. Observed-factor measurements anchor the representation, joint concept–graph dynamics expose testable future states, and a constrained decoder produces valid asymmetric return scenarios. The system-level result links admissibility, finite-horizon error propagation, and a direct downside response. The main five-seed comparison combines leading squared error with a top-two six-metric profile, while diagnostics expose the model's information use. Prediction and explanation thus share quantities whose evolution and risk effects can be evaluated.

AI USE STATEMENT

In this work, we used generative AI tools during manuscript preparation for editing and polishing. We have not used generative AI tools for dataset processing in this process, and translation and qualitative analysis are not applicable. Additionally, we used generative AI tools for manuscript organization and figure-label editing. We take responsibility for the final content of this work, including text or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The work studies predictive models using financial-market datasets and does not establish investment profitability or suitability for deployment. Reported interpretations concern the computational mechanism, not identified causal effects or individual financial recommendations. Data licensing, universe selection, market changes, and interval undercoverage limit downstream use. Any practical application requires independent validation, appropriate risk controls, and consideration of transaction costs and the consequences of prediction errors.

REPRODUCIBILITY STATEMENT

Appendix B.2 specifies point-in-time measurements and preprocessing, Appendix C gives assumptions and complete proofs, and Appendix E.2 documents splits, hyperparameters, seeds, and execution provenance. Algorithm 1 describes checkpoint selection, reloading, and final testing; the equation-to-code map links mathematical objects to implementation files. Full seed statistics, ablation provenance, scoring definitions, and additional diagnostics are provided in the experimental appendices. The comparison's source and compute-accounting boundaries are stated explicitly rather than inferred from reported metrics.

REFERENCES

Yakov Amihud. Illiquidity and stock returns: Cross-section and time-series effects. *Journal of Financial Markets*, 5(1):31–56, 2002.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.

Ole E. Barndorff-Nielsen and Neil Shephard. Econometrics of testing for jumps in financial economics using bipower variation. *Journal of Financial Econometrics*, 4(1):1–30, 2006.

Ole E. Barndorff-Nielsen, Silja Kinnebrock, and Neil Shephard. Measuring downside risk—realized semivariance. In *Volatility and Time Series Econometrics: Essays in Honor of Robert Engle*, pp. 117–136. Oxford University Press, 2010.

Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. NHITS: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6989–6997, 2023.

Ben Chamberlain, James Rowbottom, Maria I. Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. GRAND: Graph neural diffusion. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1407–1418, 2021.

Luyang Chen, Markus Pelger, and Jason Zhu. Deep learning in asset pricing. *Management Science*, 70(2):714–750, 2024.

Rama Cont. Empirical properties of asset returns: Stylized facts and statistical issues. *Quantitative Finance*, 1(2):223–236, 2001.

Rama Cont, Arseniy Kukanov, and Sasha Stoikov. The price impact of order book events. *Journal of Financial Econometrics*, 12(1):47–88, 2014.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10148–10167, 2024.

Robert Engle. Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models. *Journal of Business & Economic Statistics*, 20(3): 339–350, 2002.

Robert F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1007, 1982.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Lió, and Mateja Jamnik. Concept embedding models: Beyond the accuracy-explainability trade-off. In *Advances in Neural Information Processing Systems*, volume 35, pp. 21400–21413, 2022.

Mateo Espinosa Zarlenga, Katie Collins, Krishnamurthy Dvijotham, Adrian Weller, Zohreh Shams, and Mateja Jamnik. Learning to receive help: Intervention-aware concept embedding models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Eugene F. Fama. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2):383–417, 1970.

Eugene F. Fama and Kenneth R. French. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1):3–56, 1993.

Eugene F. Fama and Kenneth R. French. A five-factor asset pricing model. *Journal of Financial Economics*, 116(1):1–22, 2015.

Jianqing Fan, Yuan Liao, and Martina Mincheva. Large covariance estimation by thresholding principal orthogonal complements. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(4):603–680, 2013.

Kenneth R. French. Data library. Dartmouth College, 2026. Accessed 2026-09-03.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E. Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2): 243–268, 2007.

Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273, 2020.

Shihao Gu, Bryan Kelly, and Dacheng Xiu. Autoencoder asset pricing models. *Journal of Econometrics*, 222(1):429–450, 2021.

Campbell R. Harvey, Yan Liu, and Heqing Zhu. ... and the cross-section of expected returns. *The Review of Financial Studies*, 29(1):5–68, 2016.

Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. Addressing leakage in concept bottleneck models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Narasimhan Jegadeesh and Sheridan Titman. Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of Finance*, 48(1):65–91, 1993.

Jingwen Jiang, Bryan Kelly, and Dacheng Xiu. (Re-)Imag(in)ing price trends. *The Journal of Finance*, 78(6):3193–3249, 2023.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations*, 2024.

Bryan Kelly, Semyon Malamud, and Kangying Zhou. The virtue of complexity in return prediction. *The Journal of Finance*, 79(1):459–503, 2024.

Bryan T. Kelly, Seth Pruitt, and Yinan Su. Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics*, 134(3):501–524, 2019.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2668–2677, 2018.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5338–5348, 2020.

Jérôme Kunegis, Stephan Schmidt, Andreas Lommatzsch, Jürgen Lerner, Ernesto W. De Luca, and Sahin Albayrak. Spectral analysis of signed graphs for clustering, prediction and visualization. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pp. 559–570, 2010.

Tong Li, Zhaoyang Liu, Yanyan Shen, Xue Wang, Haokun Chen, and Sen Huang. MASTER: Market-guided stock transformer for stock price forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4): 1748–1764, 2021.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-XL: Long-context transformers for unified time series forecasting. In *International Conference on Learning Representations*, 2025a.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 39295–39317, 2025b.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Mazda Moayeri, Keivan Rezaei, Maziar Sanjabi, and Soheil Feizi. Text-to-concept (and back) via cross-model alignment. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 25037–25060, 2023.

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

Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *International Conference on Learning Representations*, 2023.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8857–8868, 2021.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, 2019.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191, 2020.

Ivaxi Sheth and Samira Ebrahimi Kahou. Auxiliary losses for learning generalizable concept-based models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Edgardo Solano Carrillo, Sreerag Vadakkemeppully Naveenachandran, and Julia Niebling. TEDM: Time series forecasting with elucidated diffusion models. In *International Conference on Learning Representations*, 2026.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024a.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. TimeMixer++: A general time series pattern machine for universal predictive analysis. In *International Conference on Learning Representations*, 2025a.

Yanlong Wang, Jian Xu, Tiantian Gao, Hongkang Zhang, Shao-Lun Huang, Danny Dongning Sun, and Xiao-Ping Zhang. FinTSBridge: A new evaluation suite for real-world financial prediction with advanced time series models. arXiv preprint arXiv:2503.06928, 2025b. ICLR 2025 Workshop on Advances in Financial AI.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. TimeXer: Empowering transformers for time series forecasting with exogenous variables. In *Advances in Neural Information Processing Systems*, 2024b.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53140–53164, 2024.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 753–763, 2020.

Wenyan Xu, Dawei Xiang, Yue Liu, Xiyu Wang, Yanxiang Ma, Liang Zhang, Shu Hu, Chang Xu, and Jiaheng Zhang. FinMultiTime: A four-modal bilingual dataset for financial time-series analysis. arXiv preprint arXiv:2506.05019, 2025.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *International Conference on Learning Representations*, 2023.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023.

A Extended related work

A.1 Financial measurements and learned asset-pricing representations

Observable quantities provide anchors, not sufficient statistics. Financial econometrics offers a vocabulary for distinguishing return direction, risk, liquidity, and common exposure. The autoregressive conditional-variance models of Engle (1982) and Bollerslev (1986) make volatility a forecastable state rather than an unconditional constant. Engle (2002) separately models evolving cross-series correlation. Jegadeesh & Titman (1993) document momentum strategies, while Amihud (2002) develops an illiquidity measure from absolute returns and dollar volume. Barndorff-Nielsen et al. (2010) distinguish downside variation, and Barndorff-Nielsen & Shephard (2006) develop bipower-variation-based jump testing. At a finer observational level, Cont et al. (2014) link short-interval price changes to order-flow imbalance. These are distinct measurement settings: daily signed dollar volume is not order-book imbalance, and a daily jump-share proxy does not inherit an infill-asymptotic jump test's guarantees. TREFIN adopts explicitly defined analogues where the available data permit them, including excess-return rather than total-return variation. Its multiscale vocabulary does not assume that a documented historical association yields stable incremental predictive power in our task. That question remains empirical, as does the adequacy of the selected lookback lengths.

Economic structure is compatible with expressive learning. The common-factor models of Fama & French (1993) and Fama & French (2015) motivate separating systematic exposure from residual variation. Kelly et al. (2019) make exposures conditional on characteristics through instrumental principal components, and Gu et al. (2021) extend conditional factor modeling with nonlinear autoencoders. Chen et al. (2024) use deep learning to estimate asset-pricing objects under an economic restriction, demonstrating that economically structured learning is already an established research direction. Meanwhile, Gu et al. (2020), Jiang et al. (2023), and Kelly et al. (2024) offer different evidence for the value of flexible return representations. We therefore do not position TREFIN against an allegedly unstructured asset-pricing literature. Its target differs: lead-specific return distributions with measurable intermediate state forecasts, rather than estimation of a stochastic discount factor or identification of a latent pricing factor system. In particular, its rolling observed-factor coefficients are data-estimated inputs, not latent neural factors or learned risk prices. The restriction we study is that nonlinear prediction must operate through this measured financial interface. Whether that interface discards useful information is a substantive cost to evaluate, not an assumption that economic naming can remove.

A.2 Time-series forecasting: architectural progress and task alignment

Temporal and cross-variable inductive biases. With DLinear, Zeng et al. (2023) show that simple linear forecasting models can challenge much more elaborate architectures on standard benchmarks, emphasizing the importance of strong comparisons. Nie et al. (2023) use patching and channel-independent processing in PatchTST; Liu et al. (2024) instead place attention over variate representations in iTransformer. In TimesNet, Wu et al. (2023) model temporal variation through period-dependent two-dimensional representations. Multiscale structure is developed through hierarchical interpolation in NHITS by Challu et al. (2023), decomposable mixing in TimeMixer by Wang et al. (2024a), and broader pattern extraction in TimeMixer++ by Wang et al. (2025a). These approaches offer useful biases without requiring the internal coordinates to equal measured financial quantities. Context-sensitive forecasting is also addressed directly: Wang et al. (2024b) incorporate exogenous variables in TimeXer, and Li et al. (2024) use market-guided feature selection with intra-stock and inter-stock aggregation in MASTER. The difference from TREFIN is not the presence of multiple scales, external context, or stock dependence. It is the requirement that their roles be specified through a common measurement procedure and retained in a supervised future-state path. Forecasting comparisons consequently test the cost and benefit of that additional restriction; they do not establish that generic architectures are intrinsically unsuitable for finance.

Foundation models and flexible predictive distributions. Large-scale pretraining introduces a complementary route to generalization. Das et al. (2024) develop the decoder-only forecasting foundation model TimesFM, Woo et al. (2024) design Moirai for heterogeneous series, and Ansari et al. (2024) learn probabilistic forecasting through discretized time-series tokens in Chronos. Jin et al.

(2024) instead reprogram a frozen language model in Time-LLM, while Liu et al. (2025a) develop Timer-XL's long-context formulation supporting multiple forecasting settings. These contributions must not be conflated with a task-trained reproduction of one architecture: pretraining data and transfer protocols change the question being tested. Probabilistic forecasting likewise predates our work. Salinas et al. (2020) generate forecasts with autoregressive recurrent networks in DeepAR; Rasul et al. (2021) use denoising diffusion in TimeGrad; Liu et al. (2025b) develop continuous generative forecasting with flow matching in Sundial; and Solano Carrillo et al. (2026) adapt elucidated diffusion to sequential forecasting in TEDM. Their expressive distribution families are important alternatives to a restricted parametric decoder. TREFIN pursues a different trade-off: an inspectable state interface with explicit distributional validity and selected conditional responses. It neither establishes a more expressive distribution family nor claims the zero-shot capabilities of pretrained models. Our cryptocurrency experiment is domain-specific refitting, and our implemented innovation law is cross-sectional at each lead rather than an unrestricted joint future-path law.

A.3 Interpretability, concepts, and the problem of semantic fidelity

Attribution and decomposition answer different questions. The local surrogate explanations of Ribeiro et al. (2016) and the additive feature attributions of Lundberg & Lee (2017) help analyze predictions without requiring a particular predictive architecture. Kim et al. (2018) move from low-level features to user-defined concepts through directional sensitivity in learned representations. These methods address important interpretive questions, but such analysis alone does not require a forecast to be generated by a concept state whose future evolution is supervised. In forecasting itself, Oreshkin et al. (2020) provide an interpretable basis-expansion configuration of N-BEATS, and Lim et al. (2021) combine feature selection and interpretable attention with multi-horizon prediction in Temporal Fusion Transformers. Thus neither interpretable forecasting nor structured temporal decomposition is new. Following the architectural emphasis of Rudin (2019), TREFIN adds a more specific requirement: the intermediate variables have point-in-time financial measurement rules and subsequent observations against which their predictions can be evaluated. This is operational interpretability, not a claim that every nonlinear parameter is individually understandable. Nor does changing an input concept identify an economic causal effect; the reported interventions are controlled changes to a fitted predictor under an explicitly stated computational boundary.

Concept bottlenecks require more than interpretable names. Koh et al. (2020) establish the central concept-bottleneck formulation and its potential for test-time corrections. Yuksekgonul et al. (2023) construct bottlenecks from pretrained models, and Oikarinen et al. (2023) reduce the need for manually labeled concepts. Moayeri et al. (2023) connect model representations to language-defined concepts through cross-model alignment. These developments broaden access to semantic representations, but a language-aligned or automatically discovered concept is not automatically an operational financial measurement. Espinosa Zarlenga et al. (2022) introduce richer concept embeddings to improve the accuracy–interpretability trade-off; Espinosa Zarlenga et al. (2023) explicitly train models to receive concept interventions; and Sheth & Ebrahimi Kahou (2023) study auxiliary objectives for more generalizable concept representations. TREFIN shares the principle that intermediate supervision matters, but uses continuous measured financial coordinates and supervised future trajectories rather than adopting their classification architectures. The leakage analysis of Havasi et al. (2022) remains directly relevant: soft numerical deviations can transmit unintended information. Deterministic measurement anchors our initial state, and removing a raw-history bypass enforces an information route, but finite trajectory losses do not prove semantic purity of every future prediction. Accordingly, we distinguish structural routing, empirical state fidelity, and successful downstream prediction instead of presenting them as equivalent consequences of a bottleneck.

A.4 Residual graphs, typed dynamics, and probabilistic accountability

A relationship needs an estimand before it needs a neural operator. Graph-based forecasting can learn useful relationships without specifying their financial interpretation. Wu et al. (2020), for example, jointly learn cross-variable structure and temporal prediction. For our explanatory objective, the graph instead needs an *estimand*: a stated statistical quantity that the estimation procedure targets. Friedman et al. (2008) supply sparse inverse-covariance estimation, which we apply to standardized factor residuals. Fan et al. (2013) also separate common-factor and residual covariance

Table 3: Positioning by the additional requirement studied in TREFIN. Each row describes the cited approaches' scope, not an impossibility result for an entire model family. "Inspectable" does not mean causally identified.

Research perspective	Established capability	Additional requirement in our formulation
Conditional factor learning of Gu et al. (2021)	Nonlinear, economically structured asset-pricing representations	Observable financial coordinates and separately evaluated future-state predictions
Context-aware forecasting of Li et al. (2024)	Market-guided features and stock relationships	A shared factor-adjustment definition for exposures, residual edges, and aggregate dependence
Concept bottlenecks of Koh et al. (2020)	Prediction routed through meaningful intermediate variables	Continuous, multi-scale measurements with supervised concept–graph evolution
Graph diffusion of Chamberlain et al. (2021)	Structured propagation with a dynamical interpretation	Different treatment of directional and magnitude coordinates, with code-matched finite-horizon guarantees
Generative forecasting of Rasul et al. (2021)	Multivariate predictive scenarios	Explicitly constrained downside-scale responses to named risk coordinates

structure, but use latent principal components and residual thresholding rather than our observed-factor precision construction. Neither distinction turns a residual edge into a causal link; partial correlation concerns conditional linear association, and a conditional-independence interpretation requires additional distributional assumptions. Kunegis et al. (2010) provide signed spectral operators, while Chamberlain et al. (2021) develop graph diffusion as a principled neural modeling perspective. Our contribution is the financially typed use of signed versus magnitude coupling and the supervised co-evolution of measured concepts and residual graphs. The resulting bounded nonlinear dynamics are not a literal physical diffusion process, and their admissible-state guarantee should not be confused with global contraction or recovery of the true market network.

Distribution scores, calibration, and evidence have separate roles. Gneiting & Raftery (2007) establish proper scoring rules as a principled basis for evaluating probabilistic forecasts. TREFIN uses a multivariate energy score averaged across forecast leads, together with state-path supervision and sensitivity regularization. Properness concerns the population scoring rule over probability distributions; it does not guarantee that a constrained, jointly regularized neural estimator recovers the true conditional law. Gneiting et al. (2007) further separate calibration from sharpness, motivating our coverage diagnostics alongside distributional scores. Benchmark construction also matters: the multimodal financial dataset of Xu et al. (2025) and the financial evaluation suite of Wang et al. (2025b) expand evaluation resources, but dataset availability does not make different asset panels, horizons, or target definitions comparable. Following the multiple-testing concerns raised by Harvey et al. (2016), we separate validation-based selection from chronological held-out evaluation across seeds. Table 3 locates our contribution in the *joint contract* connecting measurement, propagation, and distributional interpretation.

B Complete model specification

TREFIN is one supervised state-space model with four consecutive stages. At a *forecast origin* t—the last time whose observations may be used—a deterministic, point-in-time measurement map first converts the observed history $\mathcal{H}_t$ into three complementary states: an asset–concept matrix $\mathbf{C}_{t,0}$, a signed graph $\mathbf{A}_{t,0}$ of dependence left after observed common factors have been removed, and a market-condition vector $\mathbf{s}_t$. A learned transition then rolls the concept matrix and graph forward together; a probabilistic decoder maps every predicted state pair to a joint distribution of the N next-period asset returns; and a single objective fits the return distribution while supervising the

predicted concept and graph paths. In symbols, the computational thread is

$$\begin{aligned}
\underbrace{\mathcal{H}_t}_{\text{available history}} &\xrightarrow{\mathcal{M}} \underbrace{(\mathbf{C}_{t,0}, \mathbf{A}_{t,0}, \mathbf{s}_t)}_{\text{measured input}}, \\
(\mathbf{C}_{t,0}, \mathbf{A}_{t,0}, \mathbf{s}_t) &\xrightarrow{\Phi_{\theta,0:H-1}} \underbrace{\{\widehat{\mathbf{C}}_{t,h}, \widehat{\mathbf{A}}_{t,h}\}_{h=1}^{H}}_{\text{predicted state path}}, \\
\{\widehat{\mathbf{C}}_{t,h}, \widehat{\mathbf{A}}_{t,h}\}_{h=1}^{H} &\xrightarrow{\mathcal{D}_\theta} \underbrace{\{\hat{\mathbf{y}}_{t,h}^{(s)} : 1 \le h \le H,\ 1 \le s \le S\}}_{\text{return scenarios}}.
\end{aligned} \tag{16}$$

Here $\mathcal{M}$ contains no trainable neural parameter; $\Phi_{\theta,0:H-1}$ is the learned closed-loop state transition; $\mathcal{D}_\theta$ is the learned distribution decoder; and s indexes one of S Monte Carlo scenarios. Thus the reader can distinguish four objects throughout: quantities observed in the raw data, quantities deterministically estimated from those data, states predicted by the neural model (marked by a hat), and future measurements used only as training labels. Sections B.2–B.5 follow exactly these four stages. No raw-history skip connection bypasses the measured state interface.

B.1 Problem formulation, notation, and information boundary

Consider $N \in \mathbb{N}$ assets and a forecast horizon of $H \in \mathbb{N}$ periods. Assets are indexed by $i, j \in \{1, \dots, N\}$, concept coordinates by $k \in \{1, \dots, K\}$, and historical lags by $u \ge 0$. The symbol h always denotes a lead: return leads use $h \in \{1, \dots, H\}$, whereas state paths additionally include the observed initial state at $h = 0$. Bold lower-case and upper-case symbols denote vectors and matrices, respectively; calligraphic symbols denote sets, information collections, or named operators. For a matrix $\mathbf{X}$, $\mathbf{X}^\top$ is its transpose and $\|\mathbf{X}\|_F$ is its Frobenius norm; for a vector $\mathbf{x}$, $\|\mathbf{x}\|_2$ is its Euclidean norm and $\mathrm{diag}(\mathbf{x})$ places its entries on a diagonal. The symbols $\mathbf{I}_N$ and $\mathbf{1}_N$ denote the $N \times N$ identity matrix and the length-N all-one vector; $\odot$ denotes elementwise multiplication; and $[x]_+ = \max(x, 0)$. The sigmoid function is $\sigma(x) = (1 + \exp(-x))^{-1}$. The functions $\mathrm{softplus}(x) = \log(1 + \exp(x))$, $\mathrm{SiLU}(x) = x\sigma(x)$, and $\tanh(x)$ are applied elementwise to vector or matrix arguments.

Let $P_{t,i} > 0$ be the point-in-time price of asset i recorded for observation interval t. We distinguish the conventional total log return $g_{t,i}$ from the excess log return $r_{t,i}$ used by the repository:

$$g_{t,i} = \log\left(\frac{P_{t,i}}{P_{t-1,i}}\right), \qquad r_{t,i} = g_{t,i} - r_t^{\mathrm{rf}}. \tag{17}$$

Here r_t^{rf} is an observed, date-aligned one-interval log risk-free return, not a fitted parameter. In the principal U.S. daily data it is obtained from the Ken French Data Library by converting the reported daily simple risk-free return RF_t to $\log(1 + \mathrm{RF}_t)$; a preregistered zero series is used only for datasets, such as the cryptocurrency benchmark, whose risk-free policy is explicitly zero (French, 2026). Both the supervised labels and the asset-level concept code use $r_{t,i}$. Consequently, the volatility concepts below are precisely described as *excess-return analogues* of standard realized-variation quantities; the shift by r_t^{rf} is stated rather than conflated with the classical unshifted construction based on $g_{t,i}$. Writing $\mathbf{r}_t = (r_{t,1}, \dots, r_{t,N})^\top \in \mathbb{R}^N$, the primary supervised label at forecast origin t is

$$\mathbf{Y}_t = \begin{bmatrix} \mathbf{r}_{t+1}^\top \\ \vdots \\ \mathbf{r}_{t+H}^\top \end{bmatrix} \in \mathbb{R}^{H \times N}, \qquad [\mathbf{Y}_t]_{h,i} = r_{t+h,i}. \tag{18}$$

Thus $\mathbf{Y}_t$ is the *observed label*: the realized future excess-return path, not a model output and not an accumulated H-period return. For $H = 5$, its first row contains the next-period return of every asset and its fifth row contains every asset's fifth-period-ahead return. Returns are dimensionless decimals; for example, 0.01 represents one percent. Model-generated scenarios are always marked by a hat and introduced separately in equation (60).

The input supplied to the neural model is

$$\mathcal{I}_t = (\mathbf{C}_{t,0}, \mathbf{A}_{t,0}, \mathbf{s}_t), \quad \mathbf{C}_{t,0} \in \mathbb{R}^{N \times K}, \quad \mathbf{A}_{t,0} \in \mathbb{R}^{N \times N}, \quad \mathbf{s}_t \in \mathbb{R}^3. \tag{19}$$

Here the state-lead index 0 denotes the state measured at time t. The element $[\mathbf{C}_{t,0}]_{ik}$ is the standardized value of financial concept k for asset i. The element $[\mathbf{A}_{t,0}]_{ij}$ is the signed direct

dependence between the factor residuals of assets i and j: positive and negative values represent conditional co-movement and opposition, and diagonal entries are zero. The three entries of $\mathbf{s}_t$ are standardized market log realized variance, mean absolute factor-residual dependence, and market-weighted log illiquidity. Section B.2 gives their causal estimators and scaling rules.

Given $\mathcal{I}_t$, TREFIN with trainable parameters θ recursively predicts future concept and graph states and parameterizes the conditional path law

$$\widehat{\mathbb{P}}_\theta(\mathbf{Y}_t \mid \mathcal{I}_t) = \bigotimes_{h=1}^{H} \widehat{\mathbb{P}}_{\theta,h}\Big(\mathbf{r}_{t+h} \mid \widehat{\mathbf{C}}_{t,h}, \widehat{\mathbf{A}}_{t,h}, \mathbf{s}_t\Big), \qquad \widehat{\boldsymbol{\mu}}_{t,1:H} \in \mathbb{R}^{H\times N}, \tag{20}$$

from which return scenarios are sampled. Here $\widehat{\mathbf{C}}_{t,h}$ and $\widehat{\mathbf{A}}_{t,h}$ are the recursively predicted states, $\widehat{\mathbb{P}}_{\theta,h}$ is the joint law of the N returns at lead h, and $\bigotimes$ denotes the product of the lead-wise laws: conditional on the deterministic predicted state path, random innovations are drawn independently across leads. The entry $[\widehat{\boldsymbol{\mu}}_{t,1:H}]_{h,i}$ is the model's conditional mean for asset i at lead h. During training, the model also receives targets $\{\mathbf{C}_{t,h}, \mathbf{A}_{t,h}\}_{h=0}^{H}$, obtained by causally reapplying the same measurement rules at times $t, t+1, \ldots, t+H$. These future states are labels only: at validation or test time, the model receives only $\mathcal{I}_t$ and must generate every state with $h > 0$.

Let $\mathcal{F}_t$ denote all information available by the forecast origin. The modeling hypothesis is the approximate predictive sufficiency statement

$$\mathbb{P}(\mathbf{Y}_t \mid \mathcal{F}_t) \approx \mathbb{P}(\mathbf{Y}_t \mid \mathbf{C}_{t,0}, \mathbf{A}_{t,0}, \mathbf{s}_t). \tag{21}$$

This is a falsifiable approximation rather than a theorem: concept-path error, graph-path error, and out-of-sample distribution scores directly test whether the compressed state retains predictive information. Architecturally, the hypothesis is enforced as a strict information boundary: once $\mathcal{I}_t$ has been constructed, raw price, volume, factor, and return histories cannot enter the dynamics or decoder.

B.2 Point-in-time measurement of the predictive financial state

We now construct the three components of $\mathcal{I}_t$ using observations dated no later than t. The order is consequential: asset-level concepts are measured first; the same factor regression used to define concept exposures supplies residuals for the signed graph; and the graph's average residual dependence becomes one coordinate of the market state. Robust scaling is applied only after all three raw objects have been measured.

B.2.1 Multi-scale expert concept atlas

Let $\Lambda = \{L_1, \ldots, L_{n_L}\} \subset \mathbb{N}$ denote a predefined set of n_L concept lookback lengths, ordered such that $L_1 < \cdots < L_{n_L}$. A lookback length counts observations at the dataset's native sampling resolution: for the principal U.S. panel, $L = 5$ means five trading days, whereas for an hourly interval panel it means five hourly observations. Thus none of the following daily quantities is claimed to be a high-frequency realized-volatility estimator. The separate factor lookback L_g and market lookback L_m introduced below use the same unit of observation; in the principal daily configuration, $\Lambda = \{5, 21, 63, 126\}$, $L_g = 126$, and $L_m = 21$. We reserve the capital symbol L for a lookback length and the lower-case symbol w for a temporal weight. Specifically, for any positive L and lag $u \in \{0, \ldots, L-1\}$, the normalized weight assigned to that lag is

$$w_u^{(L)} = \frac{\exp(-u/\tau_L)}{\sum_{v=0}^{L-1} \exp(-v/\tau_L)}, \qquad \tau_L = L/3, \tag{22}$$

where $v \in \{0, \ldots, L-1\}$ is only a normalization index and $\tau_L = L/3$ is the fixed decay time associated with length L. Thus L determines how many past observations enter, whereas $w_u^{(L)}$ determines the relative contribution of observation $t-u$ within that lookback; $\sum_{u=0}^{L-1} w_u^{(L)} = 1$, and the most recent observation receives the largest weight. Let $V_{t,i}^{\$} > 0$ denote the dollar trading volume of asset i at time t. The concepts are exponentially weighted versions or transforms of established financial quantities, rather than unnamed latent coordinates: recent return continuation motivates

momentum (Jegadeesh & Titman, 1993); realized semivariance separates positive and negative variation (Barndorff-Nielsen et al., 2010); Amihud illiquidity measures price response per dollar traded (Amihud, 2002); and the jump share below is motivated by realized-variance/bipower-variation decomposition (Barndorff-Nielsen & Shephard, 2006). We first expose the unlogged quantities and only then state the transformations stored by the code.

For each $L \in \Lambda$, define weighted momentum and the positive and negative weighted realized semivariances of the excess-return series by

$$m_{t,i}^{(L)} = \sum_{u=0}^{L-1} w_u^{(L)} r_{t-u,i}, \qquad \text{weighted momentum,} \tag{23}$$

$$\mathrm{SV}_{t,i}^{+,(L)} = \sum_{u=0}^{L-1} w_u^{(L)} [r_{t-u,i}]_+^2, \qquad v_{t,i}^{+,(L)} = \log\Big(\mathrm{SV}_{t,i}^{+,(L)} + 10^{-10}\Big), \tag{24}$$

$$\mathrm{SV}_{t,i}^{-,(L)} = \sum_{u=0}^{L-1} w_u^{(L)} [-r_{t-u,i}]_+^2, \qquad v_{t,i}^{-,(L)} = \log\Big(\mathrm{SV}_{t,i}^{-,(L)} + 10^{-10}\Big), \tag{25}$$

$$\mathrm{ILLIQ}_{t,i}^{(L)} = \sum_{u=0}^{L-1} w_u^{(L)} \frac{|r_{t-u,i}|}{V_{t-u,i}^{\$} + 10^{-8}}, \qquad \ell_{t,i}^{(L)} = \log\Big(\mathrm{ILLIQ}_{t,i}^{(L)} + 10^{-18}\Big). \tag{26}$$

Here $\mathrm{SV}_{t,i}^{+,(L)}$ and $\mathrm{SV}_{t,i}^{-,(L)}$ are the recognizable unlogged semivariance pair; $v_{t,i}^{+,(L)}$ and $v_{t,i}^{-,(L)}$ are the log-transformed coordinates actually stored in the concept matrix. Likewise, $\mathrm{ILLIQ}_{t,i}^{(L)}$ is the weighted Amihud-style level and $\ell_{t,i}^{(L)}$ is its stored logarithm. The fixed constants reproduce the implementation and make the logarithms and volume division defined. Because these quantities use the excess return $r_{t,i}$ from equation (17), they are shifted analogues of the classical constructions based on total returns, not claims of algebraic identity with the latter.

The same decomposition makes the relation to total weighted variation explicit. Define weighted realized variance $\mathrm{RV}_{t,i}^{(L)}$ and normalized bipower variation $\mathrm{BV}_{t,i}^{(L)}$ by

$$\mathrm{RV}_{t,i}^{(L)} = \mathrm{SV}_{t,i}^{+,(L)} + \mathrm{SV}_{t,i}^{-,(L)} = \sum_{u=0}^{L-1} w_u^{(L)} r_{t-u,i}^2,$$

$$\mathrm{BV}_{t,i}^{(L)} = \frac{\pi}{2Z_L} \sum_{u=0}^{L-2} \sqrt{w_{u+1}^{(L)} w_u^{(L)}} |r_{t-u-1,i}|\,|r_{t-u,i}|, \qquad Z_L = \sum_{u=0}^{L-2} \sqrt{w_{u+1}^{(L)} w_u^{(L)}}, \tag{27}$$

where $Z_L > 0$ is the finite-window normalization. The resulting coordinate is

$$j_{t,i}^{(L)} = \frac{[\mathrm{RV}_{t,i}^{(L)} - \mathrm{BV}_{t,i}^{(L)}]_+}{\mathrm{RV}_{t,i}^{(L)} + 10^{-10}}. \tag{28}$$

At daily frequency, equation (28) is explicitly a coarse jump-share proxy rather than a high-frequency consistent jump estimator. Its purpose is to provide a bounded, economically named discontinuity coordinate, not to claim that daily sampling identifies quadratic-variation jumps.

We include one additional coordinate intended to represent net signed trading pressure. Its estimator depends on data availability. When only daily prices and dollar trading volumes are observed, we use the return-signed dollar-volume proxy over the longest concept lookback L_{n_L}:

$$o_{t,i}^{\mathrm{daily}} = \frac{\sum_{u=0}^{L_{n_L}-1} w_u^{(L_{n_L})} \operatorname{sign}(r_{t-u,i}) V_{t-u,i}^{\$}}{\sum_{u=0}^{L_{n_L}-1} w_u^{(L_{n_L})} V_{t-u,i}^{\$} + 10^{-12}}, \tag{29}$$

where $\operatorname{sign}(x) \in \{-1, 0, 1\}$ returns the sign of x. When interval-level order-flow imbalance $o_{t,i} \in \mathbb{R}$ is observed directly, we instead set $o_{t,i}^{\mathrm{int}} = o_{t,i}$ without applying additional smoothing. The two estimators occupy the same semantic coordinate $o_{t,i}^{\star}$ in their respective datasets; they are not assumed to be identical microstructure statistics.

B.2.2 SYSTEMATIC FACTOR EXPOSURE AND FACTOR RESIDUALS

The factor regression is the bridge between the asset-level concept matrix and the cross-asset graph. Its slopes become interpretable exposure coordinates, while its residuals remove registered common variation before direct cross-asset dependence is estimated. Let $d_f \in \mathbb{N}$ be the number of *observed* factor-return series, let $\mathbf{f}_q \in \mathbb{R}^{d_f}$ contain their values at time q, and let $L_g \in \mathbb{N}$ be the causal factor-estimation lookback length in native observation intervals. The factor values are data, not parameters chosen by TREFIN.

Concretely, the principal U.S. experiment fixes $d_f = 5$ and uses the daily Fama–French vector

$$\mathbf{f}_q = \left[f_q^{\mathrm{MKT-RF}}, f_q^{\mathrm{SMB}}, f_q^{\mathrm{HML}}, f_q^{\mathrm{RMW}}, f_q^{\mathrm{CMA}}\right]^\top. \tag{30}$$

Here MKT–RF is the market portfolio return in excess of the risk-free rate; SMB, HML, RMW, and CMA are the published small-minus-big, high-minus-low book-to-market, robust-minus-weak profitability, and conservative-minus-aggressive investment portfolio returns, respectively (Fama & French, 1993; 2015). The repository reads their date-matched numerical values from the Ken French daily five-factor file, converts reported percentages to decimal returns, and does not reconstruct or optimize the factor portfolios (French, 2026). For example, a raw file entry of 0.30 percent for SMB is converted to $f_q^{\mathrm{SMB}} = 0.003$ before regression. Other registered datasets may provide a different, explicitly named factor vector; d_f describes that input width rather than a learned model choice.

For each asset i and forecast origin t, the intercept estimator $\widehat{\alpha}_{t,i} \in \mathbb{R}$ and factor-exposure estimator $\widehat{\boldsymbol{\beta}}_{t,i} \in \mathbb{R}^{d_f}$ are defined by

$$(\widehat{\alpha}_{t,i}, \widehat{\boldsymbol{\beta}}_{t,i}) = \arg\min_{a\in\mathbb{R},\, \mathbf{b}\in\mathbb{R}^{d_f}} \left\{ \sum_{u=0}^{L_g-1} w_u^{(L_g)} \left(r_{t-u,i} - a - \mathbf{b}^\top \mathbf{f}_{t-u}\right)^2 + \lambda_\beta \|\mathbf{b}\|_2^2 \right\}, \tag{31}$$

where a and $\mathbf{b}$ are the candidate intercept and exposure vector, and $\lambda_\beta > 0$ is the pre-specified ridge penalty. Lag $u = 0$ corresponds to time t, so equation (22) makes the regression causal and gives greater weight to recent observations. The resulting $\widehat{\boldsymbol{\beta}}_{t,i}$ is re-estimated from data at every forecast origin; it is neither fixed externally nor learned by neural backpropagation.

The implementation solves all N regressions in one matrix system; Appendix C.2 gives that code-level closed form, including the unpenalized intercept and the fixed 10^{-10} numerical jitter. It is an implementation of equation (31), not a second optimization layer.

Economically, for $p \in \{1, \ldots, d_f\}$, the pth entry of $\widehat{\boldsymbol{\beta}}_{t,i}$ measures the local exposure of asset i to a one-unit return in observed factor p, conditional on the other registered factors. Retaining these slopes in the concept state makes common-risk exposure explicit (Fama & French, 2015). For each date $q \in \{t - L_g + 1, \ldots, t\}$, the corresponding factor residual is

$$\varepsilon_{q,i}^{(t)} = r_{q,i} - \widehat{\alpha}_{t,i} - \widehat{\boldsymbol{\beta}}_{t,i}^\top \mathbf{f}_q. \tag{32}$$

The superscript (t) records that the regression coefficients were estimated using the window ending at t. The residual is the part of the asset return not explained by the registered common factors within that window. Estimating the graph from these residuals, rather than from raw returns, prevents broad market co-movement from being mistaken for a direct asset-to-asset relation.

Combining the five concept families evaluated at all n_L registered lookback lengths, one order-flow coordinate, and the d_f factor exposures gives the raw concept vector

$$\begin{aligned} \widetilde{\mathbf{c}}_{t,i} = \big[&\{m_{t,i}^{(L)}\}_{L\in\Lambda}, \{v_{t,i}^{+,(L)}\}_{L\in\Lambda}, \{v_{t,i}^{-,(L)}\}_{L\in\Lambda}, \{j_{t,i}^{(L)}\}_{L\in\Lambda}, \\ &\{\ell_{t,i}^{(L)}\}_{L\in\Lambda}, o_{t,i}^{\star}, \widehat{\boldsymbol{\beta}}_{t,i}^\top\big]^\top \in \mathbb{R}^K, \qquad K = 5n_L + 1 + d_f, \end{aligned} \tag{33}$$

where $o_{t,i}^{\star}$ is either $o_{t,i}^{\mathrm{daily}}$ or $o_{t,i}^{\mathrm{int}}$, according to the data source. The raw matrix is now defined explicitly by $[\widetilde{\mathbf{C}}_{t,0}]_{i,:} = \widetilde{\mathbf{c}}_{t,i}^\top$ for every asset i. The measured neural input $\mathbf{C}_{t,0}$ is not an additional latent construction: it is exactly this raw matrix after the train-fitted coordinate transform and clipping in equation (42). This immutable coordinate order is shared by preprocessing, model masks, losses, checkpoints, and diagnostic plots. Having completed the asset-level state, we next use the residual sequence in equation (32) to construct the cross-asset state.

B.2.3 SIGNED DIRECT-RESIDUAL GRAPH

The concept matrix describes assets individually; $\mathbf{A}_{t,0}$ adds the cross-asset structure that remains after factor adjustment. The construction has one main thread: standardize the in-window factor residuals, estimate their covariance, fit a sparse precision matrix, convert it to residual partial correlations, and use those partial correlations as signed graph edges. Every object through $\mathbf{A}_{t,0}$ is calculated before neural training. To make residuals comparable across assets, define their exponentially weighted mean, scale, and standardized score within the same window by

$$\begin{aligned}
\bar{\varepsilon}_i^{(t)} &= \sum_{u=0}^{L_g-1} w_u^{(L_g)} \varepsilon_{t-u,i}^{(t)}, \\
s_i^{\varepsilon,(t)} &= \left[\sum_{u=0}^{L_g-1} w_u^{(L_g)} \big(\varepsilon_{t-u,i}^{(t)} - \bar{\varepsilon}_i^{(t)}\big)^2 + 10^{-10} \right]^{1/2} > 0, \\
z_{q,i}^{\varepsilon,(t)} &= \frac{\varepsilon_{q,i}^{(t)} - \bar{\varepsilon}_i^{(t)}}{s_i^{\varepsilon,(t)}}, \qquad \mathbf{z}_q^{\varepsilon,(t)} = (z_{q,1}^{\varepsilon,(t)}, \dots, z_{q,N}^{\varepsilon,(t)})^\top.
\end{aligned} \tag{34}$$

The fixed 10^{-10} term prevents division by zero. Standardization removes each residual series' local level and scale; the graph will therefore represent conditional dependence rather than a ranking of assets by residual volatility. The weighted residual covariance is

$$\mathbf{S}_t = \sum_{u=0}^{L_g-1} w_u^{(L_g)} \mathbf{z}_{t-u}^{\varepsilon,(t)} \mathbf{z}_{t-u}^{\varepsilon,(t)\top} + 10^{-6} \mathbf{I}_N. \tag{35}$$

For a graph-estimation penalty $\alpha_G > 0$, the Graphical Lasso estimate (Friedman et al., 2008) is

$$\widehat{\mathbf{\Theta}}_t = \arg\min_{\mathbf{\Theta} \succ \mathbf{0}} \Big\{ \operatorname{tr}(\mathbf{S}_t \mathbf{\Theta}) - \log\det \mathbf{\Theta} + \alpha_G \|\mathbf{\Theta}\|_{1,\mathrm{off}} \Big\}, \tag{36}$$

where $\mathbf{\Theta} \succ \mathbf{0}$ means positive definite, tr is the matrix trace, det is the determinant, and $\|\mathbf{\Theta}\|_{1,\mathrm{off}} = \sum_{i \neq j} |\Theta_{ij}|$ penalizes off-diagonal precision entries. The residual partial correlation and smoothly gated signed edge are

$$\rho_{t,ij}^{\varepsilon} = -\frac{\widehat{\Theta}_{t,ij}}{\sqrt{\widehat{\Theta}_{t,ii}\widehat{\Theta}_{t,jj}}}, \qquad \widetilde{A}_{t,ij} = \begin{cases} \rho_{t,ij}^{\varepsilon}\, \sigma\big((|\rho_{t,ij}^{\varepsilon}| - c_A)/\tau_A\big), & i \neq j, \\ 0, & i = j, \end{cases} \tag{37}$$

where $c_A > 0$ is the fixed soft threshold and $\tau_A > 0$ is its temperature. Positive and negative edges retain conditional co-movement and opposition, respectively; the continuous gate avoids a discontinuous top-k graph. At a graph-refit origin, $\widetilde{\mathbf{A}}_{t,0} = (\widetilde{A}_{t,ij})_{i,j=1}^N$ is the measured raw graph. Between registered refit origins, the preceding measured graph and its partial-correlation matrix are carried forward without interpolation; future estimates are never used.

The sparsity penalty α_G is a preprocessing hyperparameter, not a neural parameter and not part of the objective in equation (66). It is selected once within the purged training fold. The roles of the two standardized residuals are different: $\mathbf{z}_q^{\varepsilon,(t)}$ in equation (34) is an *in-window* observation used to estimate $\mathbf{S}_t$ and fit $\widehat{\mathbf{\Theta}}_t$; $\breve{\mathbf{z}}_{t+1}^{\varepsilon,(t)}$ is the immediately following *training-only scoring* observation used to compare candidate penalties. Each candidate α is evaluated by next-residual Gaussian negative log likelihood plus a Jaccard penalty for changes in consecutive edge supports. The selected $\widehat{\alpha}_G$ is then frozen before the neural model is trained. This is sequential train-fold hyperparameter selection, not a nested differentiable optimization inside TREFIN. Appendix C.3 states the exact criterion, tie rule, solver fallback, and refit cadence used by the repository.

B.2.4 MARKET BOUNDARY CONDITION

Whereas $\mathbf{A}_{t,0}$ retains pairwise residual relations, $\mathbf{s}_t$ summarizes the market-wide environment in which the concept–graph system will evolve. Its three raw coordinates combine aggregate volatility, the average strength of the residual relations just estimated, and aggregate illiquidity.

Let $\pi_{q,i} \geq 0$ be the point-in-time portfolio weight of asset i with $\sum_i \pi_{q,i} = 1$; when such weights are absent, the registered policy is $\pi_{q,i} = 1/N$. Recall that $g_{q,i} = \log(P_{q,i}/P_{q-1,i})$ is the total log return from equation (17). The lagged-weight market excess log return is

$$r_q^m = \log\left(1 + \sum_{i=1}^{N} \pi_{q-1,i}\big(\exp(g_{q,i}) - 1\big)\right) - r_q^{\mathrm{rf}}. \tag{38}$$

For a registered market lookback $L_m \in \Lambda$, the unstandardized market state is

$$\widetilde{\mathbf{s}}_t = \begin{bmatrix} \log\Big(\sum_{u=0}^{L_m-1} w_u^{(L_m)} (r_{t-u}^m)^2 + 10^{-12}\Big) \\ \dfrac{2}{N(N-1)} \displaystyle\sum_{1 \leq i < j \leq N} |\rho_{t,ij}^{\varepsilon}| \\ \displaystyle\sum_{i=1}^{N} \pi_{t-1,i} \ell_{t,i}^{(L_m)} \end{bmatrix} \in \mathbb{R}^3. \tag{39}$$

The three coordinates are market log realized variance, mean absolute factor-residual dependence, and market-weighted log illiquidity. The middle coordinate is deliberately based on *factor residuals*: it measures how strongly idiosyncratic components remain connected after systematic exposures have been removed, rather than measuring the mechanically high average raw correlation seen during broad market moves. During a length-H forecast, the implementation holds the observed $\mathbf{s}_t$ fixed; it is a boundary condition, not an unobserved future market-state forecast.

B.2.5 Leakage-free robust scaling and compact state sets

The preceding steps yield the raw concept matrix, residual graph, and market vector, but their coordinates have different physical units and tail behavior. We therefore fit all normalization statistics using only the measured current states at purged training forecast origins; future state labels, validation data, and test data are excluded. For each concept coordinate, the fitting sample pools that coordinate over training origins and assets; for each market coordinate, it pools over training origins. For a raw scalar coordinate x, let m_x be its training median and let a_x be its scale. The scaler uses $a_x = 1.4826\,\mathrm{median}\,|x - m_x|$ when this quantity exceeds 10^{-6}, otherwise $(Q_{0.75} - Q_{0.25})/1.349$, otherwise the standard deviation, and finally a floor of 10^{-6}; Q_p denotes the empirical training quantile of order p. It standardizes $\bar{x} = (x - m_x)/a_x$ and clips it to

$$[b_x^-, b_x^+] = [Q_{q_-}(\bar{x}) - \eta_b \Delta Q_x,\ Q_{q_+}(\bar{x}) + \eta_b \Delta Q_x], \qquad \Delta Q_x = \max\{Q_{q_+}(\bar{x}) - Q_{q_-}(\bar{x}), 10^{-6}\}, \tag{40}$$

where $0 \leq q_- < q_+ \leq 1$ are fixed projection quantiles and $\eta_b \geq 0$ is a fixed outward-margin fraction. Write $[b_k^-, b_k^+]$ for the resulting interval of concept coordinate k. Let $\mathcal{T}_C$ denote the resulting train-fitted coordinatewise standardize-and-clip map for a raw concept matrix, and let $\mathcal{T}_s$ denote the analogous separately fitted map for a raw market vector. If $[b_{s,j}^-, b_{s,j}^+]$ is the interval for market coordinate $j \in \{1,2,3\}$, then $\mathbf{s}_t$ lies in the train-defined box $\mathcal{S} = \{\mathbf{s} \in \mathbb{R}^3 : b_{s,j}^- \leq s_j \leq b_{s,j}^+ \text{ for all } j\}$. Graph entries are not coordinatewise centered or rescaled: if $B_A > 0$ is 1.05 times the registered high training quantile of $\|\widetilde{\mathbf{A}}_{t,0}\|_F$, then

$$\mathcal{A} = \{\mathbf{A} \in \mathbb{R}^{N \times N} : \mathbf{A} = \mathbf{A}^\top, \mathrm{diag}(\mathbf{A}) = \mathbf{0},\ \|\mathbf{A}\|_F \leq B_A\}, \tag{41}$$

where $\Pi_{\mathcal{A}}$ symmetrizes, zeros the diagonal, and radially projects onto the Frobenius ball. To close the measured-state notation, let $\widetilde{\mathbf{C}}_{t,h} := \widetilde{\mathbf{C}}_{t+h,0}$ and $\widetilde{\mathbf{A}}_{t,h} := \widetilde{\mathbf{A}}_{t+h,0}$ be the raw concept matrix and raw residual graph obtained by reapplying the preceding causal estimators at calendar time $t + h$. The quantities supplied as inputs or labels are exactly

$$\mathbf{C}_{t,h} = \mathcal{T}_C(\widetilde{\mathbf{C}}_{t,h}), \qquad \mathbf{A}_{t,h} = \Pi_{\mathcal{A}}(\widetilde{\mathbf{A}}_{t,h}), \qquad \mathbf{s}_t = \mathcal{T}_s(\widetilde{\mathbf{s}}_t), \quad h \in \{0, \ldots, H\}. \tag{42}$$

Here $\mathbf{C}_{t,h}$ and $\mathbf{A}_{t,h}$ are always *measurements*: at $h = 0$ they form the neural input, and at $h > 0$ they are future auxiliary labels available only during training. The learned rollout uses different symbols,

$$\widehat{\mathbf{C}}_{t,0} := \mathbf{C}_{t,0}, \qquad \widehat{\mathbf{A}}_{t,0} := \mathbf{A}_{t,0}, \tag{43}$$

and predicts $\widehat{\mathbf{C}}_{t,h}$ and $\widehat{\mathbf{A}}_{t,h}$ for every $h > 0$. Thus no future measurement enters a validation or test forecast. The maps $\mathcal{T}_C$, $\mathcal{T}_s$, the intervals, and B_A are fold-specific train estimates; none is a trainable neural parameter.

B.3 SEMANTICS-AWARE CONCEPT–GRAPH CLOSED-LOOP DYNAMICS

Starting from the anchors in equation (43), the transition must predict later states without observing future prices or volumes. For readability at one fixed forecast origin t, we suppress only that origin index and write $\widehat{\mathbf{C}}_h := \widehat{\mathbf{C}}_{t,h}$ and $\widehat{\mathbf{A}}_h := \widehat{\mathbf{A}}_{t,h}$. A hat therefore always denotes a model-generated state; the un-hatted $\mathbf{C}_{t,h}$ and $\mathbf{A}_{t,h}$ remain measured targets defined by equation (42). Let $d_e \in \mathbb{N}$ be the lead-embedding dimension, $\mathbf{e}_h \in \mathbb{R}^{d_e}$ the learned embedding of state lead h, and $\Delta > 0$ the pre-specified Euler step. Every neural map below is a three-layer multilayer perceptron (MLP): affine map, SiLU, affine map, SiLU, affine map. Each nodewise MLP shares its weights across all assets; lead dependence enters through $\mathbf{e}_h$. Consequently, jointly permuting asset rows of $\widehat{\mathbf{C}}_h$ and the corresponding rows and columns of $\widehat{\mathbf{A}}_h$ permutes the transition outputs in the same way rather than changing the transition map. The nonlinear dynamics residual additionally applies a final $\tanh$ to bound its output. These neural terms refine, but do not replace, the damping–diffusion structure defined next.

B.3.1 TYPED DIFFUSION IS DICTATED BY CONCEPT SEMANTICS

For any $\mathbf{A} \in \mathcal{A}$, define its absolute degree vector $\mathbf{d}_A = |\mathbf{A}|\mathbf{1}_N$, the signed Laplacian $\mathbf{L}_s(\mathbf{A}) = \operatorname{diag}(\mathbf{d}_A) - \mathbf{A}$, and the unsigned Laplacian $\mathbf{L}_u(\mathbf{A}) = \operatorname{diag}(\mathbf{d}_A) - |\mathbf{A}|$, where $|\mathbf{A}|$ is elementwise absolute value. Let $d_k \in \{0, 1\}$ be a fixed semantic type: $d_k = 1$ for momentum, order flow, and all factor betas, and $d_k = 0$ for both semivariances, jump share, and illiquidity. If $\mathbf{C}_{:,k}$ is column k of $\mathbf{C}$, the typed graph operator $\mathcal{L}_{\mathbf{A}} : \mathbb{R}^{N\times K} \to \mathbb{R}^{N\times K}$ is

$$\big(\mathcal{L}_{\mathbf{A}}[\mathbf{C}]\big)_{:,k} = d_k \mathbf{L}_s(\mathbf{A})\mathbf{C}_{:,k} + (1 - d_k)\mathbf{L}_u(\mathbf{A})\mathbf{C}_{:,k}. \tag{44}$$

For a negative edge, signed diffusion regards opposite directional values as agreement, while unsigned diffusion always smooths magnitudes. This distinction prevents the model from interpreting a negative edge as a request for "negative volatility" or "negative illiquidity."

Lemma 1 (Positive-semidefinite typed dissipation) *For every symmetric zero-diagonal* $\mathbf{A}$ *and every* $\mathbf{x} \in \mathbb{R}^N$*, both* $\mathbf{L}_s(\mathbf{A})$ *and* $\mathbf{L}_u(\mathbf{A})$ *are positive semidefinite. Consequently,*

$$\begin{aligned}\mathbf{x}^\top \mathbf{L}_s(\mathbf{A})\mathbf{x} &= \frac{1}{2}\sum_{i,j} |A_{ij}|(x_i - \operatorname{sign}(A_{ij})x_j)^2 \geq 0,\\ \mathbf{x}^\top \mathbf{L}_u(\mathbf{A})\mathbf{x} &= \frac{1}{2}\sum_{i,j} |A_{ij}|(x_i - x_j)^2 \geq 0,\end{aligned} \tag{45}$$

so every column selected by equation (44) *has a nonnegative diffusion energy.*

The identity follows by expanding each square and using $A_{ij} = A_{ji}$; it is also the signed-Laplacian construction studied by Kunegis et al. (2010). Lemma 1 is an algebraic guarantee and does not depend on training data.

B.3.2 PARALLEL OLD-STATE UPDATE

Let $g_\gamma : \mathbb{R}^3 \to \mathbb{R}^K$, $g_D : \mathbb{R}^3 \to \mathbb{R}$, and $g_b : \mathbb{R}^3 \to \mathbb{R}^K$ denote learned MLPs. For registered constants $0 < \gamma_{\min} < \gamma_{\max}$, $D_{\max} > 0$, and $b_{\max} > 0$, define regime-conditioned damping, diffusion, and common forcing by

$$\begin{aligned}\boldsymbol{\gamma}(\mathbf{s}_t) &= \gamma_{\min}\mathbf{1}_K + (\gamma_{\max} - \gamma_{\min})\sigma(g_\gamma(\mathbf{s}_t)) \in \mathbb{R}^K,\\ D(\mathbf{s}_t) &= D_{\max}\sigma(g_D(\mathbf{s}_t)) \in (0, D_{\max}),\\ \mathbf{b}(\mathbf{s}_t) &= b_{\max}\tanh(g_b(\mathbf{s}_t)) \in (-b_{\max}, b_{\max})^K.\end{aligned} \tag{46}$$

Let F_θ be the bounded residual MLP, let $r_F > 0$ be its fixed residual multiplier, and define the local node features

$$\mathbf{x}_{h,i} = [\widehat{\mathbf{C}}_{h,i:}^\top,\ (\widehat{\mathbf{A}}_h\widehat{\mathbf{C}}_h)_{i:}^\top,\ \mathbf{s}_t^\top,\ \mathbf{e}_h^\top]^\top \in \mathbb{R}^{2K+3+d_e}. \tag{47}$$

The learned residual row is $\mathbf{F}_{h,i:} = r_F F_\theta(\mathbf{x}_{h,i})^\top$, hence every entry of $\mathbf{F}_h \in \mathbb{R}^{N\times K}$ lies in $(-r_F, r_F)$. If $\Pi_{\mathcal{C}}$ clips coordinate k to its train-defined interval $[b_k^-, b_k^+]$, the concept step is

$$\widehat{\mathbf{C}}_{h+1} = \Pi_{\mathcal{C}}\Big[\widehat{\mathbf{C}}_h + \Delta\Big\{-\widehat{\mathbf{C}}_h \operatorname{diag}(\boldsymbol{\gamma}(\mathbf{s}_t)) - D(\mathbf{s}_t)\mathcal{L}_{\widehat{\mathbf{A}}_h}[\widehat{\mathbf{C}}_h] + \mathbf{1}_N\mathbf{b}(\mathbf{s}_t)^\top + \mathbf{F}_h\Big\}\Big]. \tag{48}$$

Damping removes stale state, typed diffusion transfers state along economically meaningful direct-residual channels, common forcing represents a regime-wide push, and the small bounded residual represents interactions not resolved by those three mechanisms. The residual is capacity-matched but amplitude-limited to control the size of the nonlinear correction relative to the damping–diffusion update.

The graph is simultaneously proposed from the *old* predicted concept state. Let U_θ and V_θ be separate learned MLP encoders into $d_g \in \mathbb{N}$ graph dimensions and define, for node input $\mathbf{q}_{h,i} = [\widehat{\mathbf{C}}_{h,i:}^\top, \mathbf{s}_t^\top, \mathbf{e}_h^\top]^\top$,

$$\mathbf{u}_{h,i} = U_\theta(\mathbf{q}_{h,i}) \in \mathbb{R}^{d_g}, \qquad \mathbf{v}_{h,i} = V_\theta(\mathbf{q}_{h,i}) \in \mathbb{R}^{d_g},$$
$$S_{h,ij} = \frac{\mathbf{u}_{h,i}^\top \mathbf{u}_{h,j} - \mathbf{v}_{h,i}^\top \mathbf{v}_{h,j}}{\sqrt{d_g}}, \qquad G_{h,ij} = \begin{cases} \tanh(S_{h,ij})\sigma((|S_{h,ij}| - c_G)/\tau_G), & i \neq j, \\ 0, & i = j, \end{cases} \tag{49}$$

where $c_G > 0$ and $\tau_G > 0$ are the future-graph gate center and temperature. The similarity Gram term creates positive affinity, the opposition Gram term creates negative affinity, and their difference is symmetric by construction. With fixed refresh $\eta_A \in (0, 1]$,

$$\widehat{\mathbf{A}}_{h+1} = \Pi_{\mathcal{A}}\big((1 - \eta_A)\widehat{\mathbf{A}}_h + \eta_A \mathbf{G}_h\big). \tag{50}$$

Equations (48) and (50) are evaluated in parallel from $(\widehat{\mathbf{C}}_h, \widehat{\mathbf{A}}_h)$; the graph proposal never reads $\widehat{\mathbf{C}}_{h+1}$. This old-state convention removes an otherwise ambiguous implicit update and is exactly the order used by the forward pass.

Together, equations (48) and (50) produce one interpretable state pair for each future lead. The decoder now turns that state trajectory into the return distribution that is scored against $\mathbf{Y}_t$.

B.4 A VALID, HEAVY-TAILED, ASYMMETRIC CROSS-SECTIONAL DECODER

After each transition, the predicted state at lead $h \in \{1, \ldots, H\}$ is decoded. The update from state $h - 1$ uses $\mathbf{e}_{h-1}$, whereas the resulting state is decoded with $\mathbf{e}_h$; this convention aligns state and return leads. The decoder separates two roles: the conditional mean may use the complete predicted state, while named risk concepts enter downside scale only through an explicit positive channel. Let $\mathcal{R} \subset \{1, \ldots, K\}$ be the fixed set of downside-semivariance, jump-share, and illiquidity coordinates. Define $m_k^{\mathcal{R}} = 1$ for $k \in \mathcal{R}$ and zero otherwise, set $\widehat{\mathbf{N}}_h = \widehat{\mathbf{A}}_h \widehat{\mathbf{C}}_h$ for the graph-aggregated predicted concept neighborhood, and form for every asset

$$\boldsymbol{\psi}_{h,i}^{\text{full}} = [\widehat{\mathbf{C}}_{h,i:}^\top, \widehat{\mathbf{N}}_{h,i:}^\top, \mathbf{s}_t^\top, \mathbf{e}_h^\top]^\top,$$
$$\boldsymbol{\psi}_{h,i}^{\text{free}} = [((\mathbf{1}_K - \mathbf{m}^{\mathcal{R}}) \odot \widehat{\mathbf{C}}_{h,i:})^\top, ((\mathbf{1}_K - \mathbf{m}^{\mathcal{R}}) \odot \widehat{\mathbf{N}}_{h,i:})^\top, \mathbf{s}_t^\top, \mathbf{e}_h^\top]^\top, \tag{51}$$

where $\mathbf{m}^{\mathcal{R}} = (m_1^{\mathcal{R}}, \ldots, m_K^{\mathcal{R}})^\top$. The "free" feature deletes direct risk-coordinate paths from the scale, precision, and tail networks; named risk concepts can affect downside scale only through the auditable nonnegative channel below. The mean is allowed to depend on the full state.

Let $a_i^r > 0$ be the train-only robust return scale of asset i, computed as 1.4826 times the training-label median absolute deviation with standard-deviation fallback and floor 10^{-4}. Let $M_\mu > 0$ be a fixed mean-bound multiplier, $a_0 > 0$ a fixed scale floor, and let f_μ, f_+, f_-, f_q be scalar-output MLPs. The structural mean, calibrated mean, and positive-shock scale are

$$\mu_{h,i}^{\text{str}} = a_i^r M_\mu \tanh(f_\mu(\boldsymbol{\psi}_{h,i}^{\text{full}})), \qquad \mu_{h,i} = \varrho\, \mu_{h,i}^{\text{str}}, \tag{52}$$
$$\sigma_{h,i}^+ = a_i^r \left(a_0 + \text{softplus}(f_+(\boldsymbol{\psi}_{h,i}^{\text{free}}))\right) > 0, \tag{53}$$

At fixed origin t, the entry in the mean array from equation (20) is $[\widehat{\boldsymbol{\mu}}_{t,1:H}]_{h,i} = \mu_{h,i}$. The scalar $\varrho \in (0, 1]$ equals one during neural training and can later be selected on validation data only as described below. Let $\mathbf{a}^{\mathcal{R}} \in \mathbb{R}^{|\mathcal{R}|}$ be trainable logits, let $a_s \in \mathbb{R}$ be a trainable scalar, and let $R_{\max} > 0$ be a fixed maximum risk strength. The learned risk slopes are

$$w_k^{\mathcal{R}} = R_{\max}\sigma(a_s)\frac{\exp(a_k^{\mathcal{R}})}{\sum_{q \in \mathcal{R}} \exp(a_q^{\mathcal{R}})} > 0, \qquad k \in \mathcal{R}, \tag{54}$$

where q in the denominator is a risk-coordinate index local to this equation. Their sum is bounded above by $R_{\max}$. The negative-shock scale is

$$\sigma^{-}_{h,i} = a^{r}_{i}\left[a_0 + \mathrm{softplus}\left(f_{-}(\boldsymbol{\psi}^{\mathrm{free}}_{h,i}) + \sum_{k\in\mathcal{R}} w^{\mathcal{R}}_{k}\widehat{C}_{h,ik}\right)\right] > 0. \tag{55}$$

Holding all other coordinates, the graph, market state, and lead fixed,

$$\frac{\partial\sigma^{-}_{h,i}}{\partial\widehat{C}_{h,ik}} = a^{r}_{i}w^{\mathcal{R}}_{k}\,\sigma\left(f_{-}(\boldsymbol{\psi}^{\mathrm{free}}_{h,i}) + \sum_{q\in\mathcal{R}} w^{\mathcal{R}}_{q}\widehat{C}_{h,iq}\right) > 0 \quad (k\in\mathcal{R}). \tag{56}$$

Thus named risk acts monotonically on conditional downside scale by parameterization, not by a sampled derivative heuristic. Because $\widehat{C}_{h,ik}$ is standardized, the derivative is with respect to one robust-scale unit of the predicted named concept rather than one raw physical unit. Equation (56) is a *direct decoder* guarantee; it does not assert that a closed-loop intervention remains monotone after other concepts and the graph react.

To form a valid cross-sectional dependence law, define

$$\begin{aligned}
q_{h,i} &= \epsilon_Q + \mathrm{softplus}(f_q(\boldsymbol{\psi}^{\mathrm{free}}_{h,i})) > \epsilon_Q,\\
\lambda_{Q,h} &= 0.01 + \mathrm{softplus}\left(f_\lambda\left(\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\psi}^{\mathrm{free}}_{h,i}\right)\right) > 0,\\
\nu_h &= \nu_{\min} + \mathrm{softplus}\left(f_\nu\left(\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\psi}^{\mathrm{free}}_{h,i}\right)\right) > \nu_{\min} > 2,
\end{aligned} \tag{57}$$

where $\epsilon_Q > 0$ and $\nu_{\min} > 2$ are fixed constants and f_λ, f_ν are scalar MLPs. The graph-structured precision, covariance, and correlation are

$$\begin{aligned}
&\mathbf{Q}_h = \mathrm{diag}(\mathbf{q}_h) + \lambda_{Q,h}\mathbf{L}_s(\widehat{\mathbf{A}}_h) + \epsilon_Q\mathbf{I}_N, \qquad && \boldsymbol{\Sigma}_h = \mathbf{Q}_h^{-1},\\
&\mathbf{D}^{\Sigma}_h = \mathrm{diag}(\Sigma_{h,11},\ldots,\Sigma_{h,NN}), && \mathbf{R}_h = (\mathbf{D}^{\Sigma}_h)^{-1/2}\boldsymbol{\Sigma}_h(\mathbf{D}^{\Sigma}_h)^{-1/2} + \epsilon_R\mathbf{I}_N,
\end{aligned} \tag{58}$$

where $\mathbf{q}_h = (q_{h,1},\ldots,q_{h,N})^\top$, $\mathbf{D}^{\Sigma}_h$ contains the marginal variances, and the fixed jitter is $\epsilon_R = 10^{-5}$. The diagonal $q_{h,i}$ is an asset-specific baseline precision (larger values suppress conditional variance), while $\lambda_{Q,h}$ controls how strongly the signed residual graph couples the assets. Lemma 1 and the diagonal floors make $\mathbf{Q}_h$ strictly positive definite for every admissible graph, hence inversion and normalization are always defined. The matrix before the $\epsilon_R\mathbf{I}_N$ term is a correlation matrix; $\mathbf{R}_h$ is its explicitly jittered numerical version. Scenario generation uses the Cholesky factor $\mathbf{J}_h$ satisfying $\mathbf{J}_h\mathbf{J}_h^\top = \mathbf{R}_h + \epsilon_R\mathbf{I}_N$; the second identical jitter is an explicit numerical safeguard in the code.

For scenario $s \in \{1,\ldots,S\}$, draw $\mathbf{g}^{(s)}_h \sim \mathcal{N}(\mathbf{0},\mathbf{I}_N)$ and $\chi^{(s)}_h \sim \chi^2_{\nu_h}$ independently across scenarios and leads, set

$$\boldsymbol{\xi}^{(s)}_h = \sqrt{\frac{\nu_h - 2}{\chi^{(s)}_h}}\,\mathbf{J}_h\mathbf{g}^{(s)}_h, \qquad \kappa(\nu) = \frac{\sqrt{\nu-2}\,\Gamma((\nu-1)/2)}{2\sqrt{\pi}\,\Gamma(\nu/2)}, \tag{59}$$

where $\mathcal{N}$ is the multivariate normal law, χ^2_ν is chi-square with ν degrees of freedom, and Γ is the gamma function. Apart from the tiny registered correlation jitter, each marginal of $\boldsymbol{\xi}^{(s)}_h$ has unit variance. The final asymmetric heavy-tailed return scenario is

$$\widehat{y}^{(s)}_{t,h,i} = \mu_{h,i} + \sigma^{+}_{h,i}[\xi^{(s)}_{h,i}]_+ - \sigma^{-}_{h,i}[-\xi^{(s)}_{h,i}]_+ - \kappa(\nu_h)(\sigma^{+}_{h,i} - \sigma^{-}_{h,i}). \tag{60}$$

Here $\widehat{y}^{(s)}_{t,h,i}$ is the model-generated scenario-s excess return for asset i at calendar time $t{+}h$, conditional on the input measured at origin t; it is the scalar entry stacked into the scenario tensors used by the loss below. It must not be confused with the observed label $r_{t+h,i}$ in equation (18). Because the latent law is symmetric, $\mathbb{E}[\xi_{h,i}] = 0$ and $\mathbb{E}[[\xi_{h,i}]_+] = \mathbb{E}[[-\xi_{h,i}]_+] = \kappa(\nu_h)$, where $\mathbb{E}$ denotes expectation. Therefore $\mathbb{E}[\widehat{y}^{(s)}_{t,h,i} \mid \mathcal{I}_t] = \mu_{h,i}$ exactly in the ideal unjittered law: asymmetry cannot silently shift the named conditional mean. Innovations share one chi-square scale across assets within a lead, creating joint extremes, while the current implementation draws different leads independently conditional on their recursively coupled states. Accordingly, the decoder is a joint *cross-sectional* distribution at every lead; it does not claim to identify an unrestricted cross-horizon innovation copula.

B.5 JOINT SUPERVISION WITH A PROPER DISTRIBUTION OBJECTIVE

Training must identify both the predictive return law and the financial meaning of its internal trajectory. The first requirement is enforced by a proper multivariate score on sampled returns; the second is enforced by supervised concept and graph paths. Consider a minibatch containing $B \in \mathbb{N}$ forecast origins $t_1, \dots, t_B$. For batch item $b \in \{1, \dots, B\}$ and lead $h \in \{1, \dots, H\}$, define the observed label vector and generated scenario vector componentwise by

$$\mathbf{y}_{b,h} := \mathbf{r}_{t_b+h}, \quad [\mathbf{y}_{b,h}]_i = r_{t_b+h,i}, \qquad [\widehat{\mathbf{y}}^{(s)}_{b,h}]_i := \widehat{y}^{(s)}_{t_b,h,i}, \quad i \in \{1, \dots, N\},\ s \in \{1, \dots, S\}. \tag{61}$$

Thus $\mathbf{y}_{b,h} \in \mathbb{R}^N$ is read from the future-return array in equation (18), while $\widehat{\mathbf{y}}^{(s)}_{b,h} \in \mathbb{R}^N$ is produced by the transition and equation (60); neither symbol is an unspecified target or output. We require $S \geq 2$, and stacking all generated vectors gives the code-level layout $[S, B, H, N]$. The exact finite-sample energy score is

$$\mathcal{L}_{\mathrm{ES}} = \frac{1}{BH} \sum_{b=1}^{B} \sum_{h=1}^{H} \left[\frac{1}{S} \sum_{s=1}^{S} \|\widehat{\mathbf{y}}^{(s)}_{b,h} - \mathbf{y}_{b,h}\|_2 - \frac{1}{2S(S-1)} \sum_{\substack{s,s'=1 \\ s \neq s'}}^{S} \|\widehat{\mathbf{y}}^{(s)}_{b,h} - \widehat{\mathbf{y}}^{(s')}_{b,h}\|_2 \right]. \tag{62}$$

All ordered unequal scenario pairs are included; memory chunking changes neither sum nor denominator. The energy score is a proper multivariate scoring rule (Gneiting & Raftery, 2007): its first term rewards calibration around the realization and its second term prevents distributional collapse. Consistent with the decoder boundary, equation (62) is a cross-sectional score averaged over leads, not a score on a flattened HN-dimensional temporal copula.

Future-state supervision penalizes deviations from the measured concept and graph targets. Define the unit-threshold Huber function

$$\mathrm{Huber}(x) = \begin{cases} \frac{1}{2}x^2, & |x| < 1, \\ |x| - \frac{1}{2}, & |x| \geq 1. \end{cases} \tag{63}$$

Let $a^C_k > 0$ be the train-only scale $\max\{1.4826\,\mathrm{median}\,|C_{t,h+1,ik} - C_{t,h,ik}|, 10^{-3}\}$ over training paths, and let $a^A > 0$ be the analogous off-diagonal graph-increment scale floored at 10^{-4}. For predictions $\widehat{\mathbf{C}}_{b,h}$ and $\widehat{\mathbf{A}}_{b,h}$, including the anchored input at $h = 0$, the auxiliary losses are

$$\mathcal{L}_C = \frac{1}{B(H+1)NK} \sum_{b=1}^{B} \sum_{h=0}^{H} \sum_{i=1}^{N} \sum_{k=1}^{K} \mathrm{Huber}\left(\frac{\widehat{C}_{b,h,ik} - C_{b,h,ik}}{a^C_k} \right), \tag{64}$$

$$\mathcal{L}_A = \frac{1}{B(H+1)N(N-1)} \sum_{b=1}^{B} \sum_{h=0}^{H} \sum_{i \neq j} \mathrm{Huber}\left(\frac{\widehat{A}_{b,h,ij} - A_{b,h,ij}}{a^A} \right). \tag{65}$$

The main model uses all K concept coordinates. In the registered no-risk ablation only, a fixed active-coordinate mask removes risk coordinates from both the state and the denominator of equation (64). The $h = 0$ terms verify that input sanitization has preserved the measured initial state; they contain no future information.

Let $\widehat{\rho}_B \geq 0$ denote the differentiable state-conditional small-gain radius defined in Section B.6, let $\rho_\star \in (0, 1)$ be a fixed training target, and define $\mathcal{L}_{\mathrm{gain}} = [\widehat{\rho}_B - \rho_\star]^2_+$. The risk slope audit is $\mathcal{L}_{\mathrm{mono}} = |\mathcal{R}|^{-1} \sum_{k \in \mathcal{R}} [-w^{\mathcal{R}}_k]^2_+$, which is identically zero up to numerical error because equation (54) is positive. Let θ collect exactly the parameters updated by backpropagation: all weights and biases of $g_\gamma, g_D, g_b, F_\theta, U_\theta, V_\theta, f_\mu, f_+, f_-, f_q, f_\lambda, f_\nu$, all lead embeddings $\{\mathbf{e}_h\}_{h=0}^{H}$, and the risk logits $(\mathbf{a}^{\mathcal{R}}, a_s)$. With fixed nonnegative loss weights $\lambda_C, \lambda_A, \lambda_M, \lambda_G$, the optimized objective is

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{ES}} + \lambda_C \mathcal{L}_C + \lambda_A \mathcal{L}_A + \lambda_M \mathcal{L}_{\mathrm{mono}} + \lambda_G \mathcal{L}_{\mathrm{gain}}. \tag{66}$$

This is joint supervision of one state-space model: the proper score trains the predictive return law, while the auxiliary paths penalize its internal state for drifting away from the financial quantities that make that law auditable. The return labels, future measured-state labels, rolling factor coefficients, residual graphs, scalers, projection bounds, selected $\widehat{\alpha}_G$, and loss weights are held fixed while minimizing $\mathcal{L}(\theta)$; they are not entries of θ. The optional calibration scalar ϱ is also excluded from this optimization and is selected only after checkpoint selection.

B.5.1 VALIDATION-ONLY PREDICTABLE-SIGNAL CALIBRATION

After selecting the best neural checkpoint, the optional calibrated configuration adjusts only the predictable mean. Write every stored validation scenario as $\widehat{\mathbf{y}}^{(s)} = \boldsymbol{\mu}^{\mathrm{str}} + \boldsymbol{\epsilon}^{(s)}$, where $\boldsymbol{\epsilon}^{(s)}$ is its realized scenario innovation. For $\varrho \in [\varrho_{\min}, 1]$ with fixed $\varrho_{\min} = 0.05$, calibration returns

$$\widehat{\mathbf{y}}^{(s)}(\varrho) = \varrho\boldsymbol{\mu}^{\mathrm{str}} + \boldsymbol{\epsilon}^{(s)}. \tag{67}$$

Every pairwise difference $\widehat{\mathbf{y}}^{(s)}(\varrho) - \widehat{\mathbf{y}}^{(s')}(\varrho)$ is invariant in ϱ, while the first term of equation (62) is a sum of Euclidean norms of affine functions of ϱ and is therefore convex. Bounded scalar minimization, including both endpoints as candidates, obtains the global validation optimum to tolerance 10^{-5}. Strict positivity preserves the sign of every nonzero structural mean. Validation calibration occurs after checkpoint selection and never inspects test outcomes.

B.6 SYSTEM-LEVEL VALIDITY AND FINITE-HORIZON SENSITIVITY

The measurement, transition, decoder, and objective now form one closed system. Its theoretical role is to provide a valid, auditable finite-horizon forecast, not to impose asymptotic convergence of financial states. The theorem below combines invariant state sets, valid probability parameters, finite-horizon error propagation, and a conditional risk response. It does not assume that every state perturbation shrinks. For the statement, define the train-derived concept box

$$\mathcal{C} = \{\mathbf{C} \in \mathbb{R}^{N\times K} : b_k^- \le C_{ik} \le b_k^+ \text{ for every } i, k\} \tag{68}$$

from the coordinate intervals in equation (40). Let $\Phi_h(\mathbf{C}, \mathbf{A}; \mathbf{s})$ denote the joint map in equations (48)–(50) at lead h, for a fixed admissible market state $\mathbf{s} \in \mathcal{S}$. Suppose nonnegative constants $m_{CC}, m_{CA}, m_{AC}, m_{AA}$ simultaneously upper-bound the four state-map blocks so that, for any two admissible states $(\mathbf{C}, \mathbf{A})$ and $(\mathbf{C}', \mathbf{A}')$,

$$\begin{bmatrix} \|\Phi_h^C(\mathbf{C}, \mathbf{A}; \mathbf{s}) - \Phi_h^C(\mathbf{C}', \mathbf{A}'; \mathbf{s})\|_F \\ \|\Phi_h^A(\mathbf{C}, \mathbf{A}; \mathbf{s}) - \Phi_h^A(\mathbf{C}', \mathbf{A}'; \mathbf{s})\|_F \end{bmatrix} \le \mathbf{M} \begin{bmatrix} \|\mathbf{C} - \mathbf{C}'\|_F \\ \|\mathbf{A} - \mathbf{A}'\|_F \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} m_{CC} & m_{CA} \\ m_{AC} & m_{AA} \end{bmatrix}, \tag{69}$$

where Φ_h^C and Φ_h^A are the concept and graph outputs, the vector inequality is componentwise, and $\mathbf{M} \in \mathbb{R}_+^{2\times 2}$ is a nonnegative block sensitivity matrix, uniform over the registered states and the H lead maps. Such finite constants exist for the fixed finite-weight networks on the compact state sets; explicit conservative choices are given in Appendix C. The spectral-radius summary retained by the implementation is

$$\rho(\mathbf{M}) = \frac{1}{2}\left[m_{CC} + m_{AA} + \sqrt{(m_{CC} - m_{AA})^2 + 4m_{CA}m_{AC}}\right]. \tag{70}$$

No condition that this radius be below one is required in the following result.

Theorem 2 (TREFIN validity and finite-horizon error propagation) *Fix trained parameters and a market state $\mathbf{s} \in \mathcal{S}$. Then:*

(i) ***State and distribution validity.*** *Starting from $(\mathbf{C}_0, \mathbf{A}_0) \in \mathcal{C} \times \mathcal{A}$, every predicted state remains in $\mathcal{C} \times \mathcal{A}$. At every lead, $\mathbf{Q}_h$ is strictly positive definite, $\sigma_{h,i}^+$ and $\sigma_{h,i}^-$ are strictly positive, $\nu_h > 2$, and equation* (60) *has a finite first moment equal to $\mu_{h,i}$ in the ideal unjittered law.*

(ii) ***Finite-horizon sensitivity.*** *For two trajectories sharing the fixed market state and lead schedule, define $\boldsymbol{\delta}_h = [\|\mathbf{C}_h - \mathbf{C}'_h\|_F, \|\mathbf{A}_h - \mathbf{A}'_h\|_F]^\top \in \mathbb{R}_+^2$, their concept/graph error vector at lead h. If both trajectories follow the learned transition, then*

$$\boldsymbol{\delta}_h \le \mathbf{M}^h \boldsymbol{\delta}_0, \qquad h \in \{1, \dots, H\}. \tag{71}$$

More generally, suppose the primed trajectory is an admissible reference path. At step $q \in \{0, \dots, H-1\}$, let $\boldsymbol{\varepsilon}_q = [\varepsilon_{C,q}, \varepsilon_{A,q}]^\top \in \mathbb{R}_+^2$ bound the Frobenius distances between its next state and the learned transition evaluated at its current state, separately for concepts

and graph. Define the state-set diameter vector $\mathbf{d} = [(N\sum_{k=1}^{K}(b_k^+ - b_k^-)^2)^{1/2},\, 2B_A]^\top$. *Then*

$$\boldsymbol{\delta}_h \le \min\left\{\mathbf{d},\ \mathbf{M}^h\boldsymbol{\delta}_0 + \sum_{q=0}^{h-1}\mathbf{M}^{h-1-q}\boldsymbol{\varepsilon}_q\right\}, \tag{72}$$

where the minimum is componentwise and $\mathbf{M}^0$ *is the* 2×2 *identity. Thus the bound separates measurement/initialization error from finite-step transition error; it is not a claim that either error decays. For an optional distributional extension, let* W_1 *denote first Wasserstein distance (the minimum expected Euclidean distance over couplings of two laws). Assume that the fixed decoder has a finite Lipschitz constant* $L_D \ge 0$ *in* W_1 *under the sum of the two state-block norms, uniformly over the registered leads. Let* $\varepsilon_D \ge 0$ *bound its distributional error at the reference states. Writing* $\widehat{\mathbb{P}}_h$ *for the predicted law,* $\mathbb{P}_h^\star$ *for the reference law, and* $\mathbf{1}_2 = (1,1)^\top$, *we additionally have*

$$W_1(\widehat{\mathbb{P}}_h, \mathbb{P}_h^\star) \le L_D\mathbf{1}_2^\top\boldsymbol{\delta}_h + \varepsilon_D, \tag{73}$$

This extension explicitly requires decoder regularity and a reference-law defect bound; the finite-horizon state bound itself does not.

(iii) ***Direct risk monotonicity and safe calibration.*** *Holding* $\widehat{\mathbf{A}}_h$, $\mathbf{s}_t$, $\mathbf{e}_h$, *and every entry of* $\widehat{\mathbf{C}}_h$ *except one* $\widehat{C}_{h,ik}$ *with* $k \in \mathcal{R}$ *fixed, downside scale is strictly increasing in that entry as in equation* (56). *The validation objective induced by equation* (67) *is convex; its pairwise energy term, tail innovations, and scenario dispersion are unchanged, and* $\varrho > 0$ *preserves mean-forecast sign.*

Proof sketch. Coordinate clipping and $\Pi_{\mathcal{A}}$ are Euclidean projections, proving state invariance. Lemma 1, $q_{h,i} > \epsilon_Q$, and the added $\epsilon_Q\mathbf{I}_N$ imply $\lambda_{\min}(\mathbf{Q}_h) > 0$; softplus and $\nu_{\min} > 2$ give the remaining distribution conditions. Symmetry of the standardized Student law gives the mean correction in equation (60). Iteration of the componentwise recurrence $\boldsymbol{\delta}_{q+1} \le \mathbf{M}\boldsymbol{\delta}_q + \boldsymbol{\varepsilon}_q$ gives the finite sum in equation (72); the diameter of the invariant set supplies its second, independent bound. The stated decoder Lipschitz premise and the triangle inequality yield equation (73). Differentiating softplus proves direct monotonicity. Finally, a norm of an affine function is convex and the common translation in equation (67) cancels from every scenario pair. Appendix C gives the code-matched constants and full argument.

The repository retains a conservative structural sensitivity matrix, a separate state-conditional training regularizer, and local Jacobian-vector/vector-Jacobian diagnostics. These are three different quantities. The regularizer encourages smaller sensitivity without hard-constraining it; local power-iteration estimates describe sampled states and are not guaranteed upper bounds. The theorem supplies a finite-horizon decomposition and invariant feasibility, not global contraction, optimal prediction, or a guarantee that the observed defects are small.

B.7 Leakage-free optimization, selection, and final prediction

The theorem concerns a fixed fitted model; we finally specify how that model is fitted and evaluated without allowing information to cross split boundaries. Forecast origins are ordered chronologically. The maximum lookback determines the first usable origin and H determines the last. Train, validation, and test blocks are separated by a purge/embargo of at least H observations, and the implementation verifies that the last training label precedes the first validation origin and the last validation label precedes the first test origin. Graph-penalty selection, robust scalers, compact bounds, label scales, and return scales use only purged training origins.

Neural parameters are optimized with AdamW, cosine learning-rate annealing, and global gradient-norm clipping. Each epoch logs post-update train and validation energy score, MSE, RMSE, MAE, and loss components under fixed random numbers; no per-epoch function accepts the test split. The best checkpoint minimizes validation energy score, with configured minimum improvement and early-stopping patience. If enabled, ϱ is then fitted once on validation scenarios. A fresh model instance reconstructs all data-derived buffers and configuration, reloads the best checkpoint, and must reproduce its validation mean to maximum absolute error below 10^{-7}. Only after that equality check is the final test set evaluated once, using 256 scenarios in the principal real-data configuration. MAPE is excluded because excess returns frequently cross or approach zero.

Algorithm 1 TREFIN training and final-test protocol at one chronological fold

Require: Point-in-time balanced panel; factors; registered windows, split boundaries, and hyperparameters.

1: Form excess-return labels by equation (18); purge and embargo all split boundaries.
2: On training origins only, select α_G using equation (79); fit robust scalers, coordinate boxes, B_A, a_i^r, a_k^C, and a^A.
3: **for** each causal origin t in chronological preprocessing order **do**
4: Refit equation (31); measure equation (33) and equation (39).
5: On the registered cadence, estimate equations (36)–(37); otherwise carry the preceding graph forward.
6: Store input $(\mathbf{C}_{t,0}, \mathbf{A}_{t,0}, \mathbf{s}_t)$, future return label $\mathbf{Y}_t$, and auxiliary paths $\{\mathbf{C}_{t,h}, \mathbf{A}_{t,h}\}_{h=0}^{H}$.
7: **end for**
8: **for** each training epoch **do**
9: **for** each shuffled training minibatch **do**
10: Roll equations (48) and (50) for H steps; sample equation (60).
11: Differentiate equation (66); clip the gradient norm; take one AdamW step.
12: **end for**
13: Evaluate fixed-random-number train and validation metrics; save best and last checkpoints; apply validation early stopping.
14: **end for**
15: Reload the best checkpoint; optionally solve the validation-only convex calibration in equation (67); save it back to the checkpoint.
16: Instantiate a fresh model, reload the saved checkpoint, and require validation-mean equality below 10^{-7}.
17: Access the test block once; save means, distribution parameters, $S_{\text{test}} = 256$ scenarios, metrics, invariant checks, and structural/local certificate reports.

What is fixed and what is learned. The concept formulas, concept ordering and types, factor list, window lengths, ridge and graph candidate grids, graph refit cadence, robust-scaling rules, split boundaries, architecture widths, projection rules, positivity floors, and loss weights are experiment configuration or train-fold estimates. Rolling betas, factor residuals, measured concepts, measured graphs, market states, scalers, and bounds are deterministic functions of the point-in-time data once that configuration is fixed. Learned by backpropagation are all MLP weights and biases, lead embeddings, similarity/opposition encoders, decoder heads, and the risk logits $(\mathbf{a}^{\mathcal{R}}, a_s)$. The optional scalar ϱ is learned only by one-dimensional validation calibration after neural training. This separation is important: factor beta is data-estimated at each origin; it is not a neural parameter.

Implementation scope. The primary loader requires a balanced panel and rejects duplicate asset–time rows, nonpositive prices or dollar volumes, missing factor observations, and any recorded availability time after its observation time. Therefore the present Methodology does not claim a learned missingness imputer. The current FinMultiTime U.S. cohort is a fixed training-era liquidity-selected set, not a point-in-time survivorship-free S&P 500 universe. Daily order flow and jump share remain proxies, and aggregate interval imbalance is not a limit-order-book representation. These boundaries are part of the model definition rather than post-hoc caveats.

C Derivations and full proofs

This appendix makes the correspondence between Theorem 2 and the executable implementation explicit. The result concerns invariant feasibility, finite-horizon error propagation, and conditional decoder responses; it does not assert global contraction or optimality of a fitted forecast.

C.1 Tensor contract and forecast timing

For one minibatch, the input tensors have shapes

$$\mathbf{C}_0 : [B, N, K], \qquad \mathbf{A}_0 : [B, N, N], \qquad \mathbf{s} : [B, 3]. \tag{74}$$

The recurrent outputs have shapes $\widehat{\mathbf{C}}_{1:H} : [B, H, N, K]$ and $\widehat{\mathbf{A}}_{1:H} : [B, H, N, N]$; after prepending the sanitized input, the supervised paths are $[B, H+1, N, K]$ and $[B, H+1, N, N]$. Return means and distribution parameters use $[B, H, N]$, cross-sectional matrices use $[B, H, N, N]$, and sampled returns use $[S, B, H, N]$. At update index $h \in \{0, \dots, H-1\}$, the code applies $\mathbf{e}_h$ to equations (48) and (50), then applies $\mathbf{e}_{h+1}$ to the decoder. All graph and concept updates read the same old state; in particular, the graph proposal does not read the newly projected concept state.

The label path is not an accumulated return. For each origin t, the stored array satisfies

$$[\mathbf{Y}_t]_{h,i} = r_{t+h,i}, \qquad h \in \{1, \dots, H\}, \tag{75}$$

and is supervised jointly across assets at each lead. The auxiliary target paths are causally recomputed measured states at $t, t+1, \dots, t+H$. They are training labels only; at prediction time the only state supplied is the measured state at t.

C.2 Matrix implementation of the rolling factor fit

This subsection gives the code-level matrix solution of equation (31); it introduces no additional statistical model. Let $q_\ell = t - L_g + \ell$ be the chronological time in row $\ell \in \{1, \dots, L_g\}$ and define

$$[\mathbf{X}_t]_{\ell,:} = \begin{bmatrix} 1 & \mathbf{f}_{q_\ell}^\top \end{bmatrix}, \qquad [\mathbf{D}_t^{\mathrm{lag}}]_{\ell,\ell} = w_{L_g-\ell}^{(L_g)}, \qquad [\mathbf{R}_t^{\mathrm{win}}]_{\ell,i} = r_{q_\ell,i}. \tag{76}$$

Here $\mathbf{X}_t \in \mathbb{R}^{L_g \times (1+d_f)}$ is the factor design matrix including its intercept column, $\mathbf{D}_t^{\mathrm{lag}} \in \mathbb{R}^{L_g \times L_g}$ is the diagonal recency-weight matrix, and $\mathbf{R}_t^{\mathrm{win}} \in \mathbb{R}^{L_g \times N}$ contains the excess returns of all N assets on the same dates. With $\mathbf{D}_f = \mathrm{diag}(0, 1, \dots, 1) \in \mathbb{R}^{(1+d_f)\times(1+d_f)}$, whose leading zero leaves the intercept unpenalized, the implementation solves

$$\widehat{\mathbf{B}}_t = (\mathbf{X}_t^\top \mathbf{D}_t^{\mathrm{lag}} \mathbf{X}_t + \lambda_\beta \mathbf{D}_f + 10^{-10} \mathbf{I}_{1+d_f})^{-1} \mathbf{X}_t^\top \mathbf{D}_t^{\mathrm{lag}} \mathbf{R}_t^{\mathrm{win}}, \qquad [\widehat{\mathbf{B}}_t]_{:,i} = \begin{bmatrix} \widehat{\alpha}_{t,i} \\ \widehat{\boldsymbol{\beta}}_{t,i} \end{bmatrix}. \tag{77}$$

The matrix $\mathbf{I}_{1+d_f}$ is the identity of order $1 + d_f$, and 10^{-10} is the fixed numerical jitter used by the repository. Consequently, column i of $\widehat{\mathbf{B}}_t$ is exactly the weighted ridge solution for asset i in equation (31); the matrix form merely evaluates all assets together.

C.3 Train-only selection of the measured residual graph

Let $\mathcal{G}_A \subset \mathbb{R}_{>0}$ be the ordered, registered candidate list for the Graphical-Lasso penalty and let $\mathcal{T}_A$ be the selected set of scoring origins inside the purged training fold. For candidate $\alpha \in \mathcal{G}_A$, write $\widehat{\boldsymbol{\Theta}}_t(\alpha)$ and $\rho_{t,ij}^{\varepsilon}(\alpha)$ for equations (36) and (37) evaluated with that candidate. The immediately following standardized residual used only for candidate scoring is

$$\breve{z}_{t+1,i}^{\varepsilon,(t)} = \frac{r_{t+1,i} - \widehat{\alpha}_{t,i} - \widehat{\boldsymbol{\beta}}_{t,i}^\top \mathbf{f}_{t+1}}{s_i^{\varepsilon,(t)}}, \qquad \breve{\mathbf{z}}_{t+1}^{\varepsilon,(t)} = (\breve{z}_{t+1,1}^{\varepsilon,(t)}, \dots, \breve{z}_{t+1,N}^{\varepsilon,(t)})^\top. \tag{78}$$

The breve distinguishes this next-period training score from the in-window $\mathbf{z}_q^{\varepsilon,(t)}$ used to fit the covariance. Equation (78) deliberately divides by the fitted in-window scale without subtracting the fitted residual mean a second time, matching the repository. Define the candidate edge support $\mathcal{E}_t(\alpha) = \{(i,j) : i \neq j,\ |\rho_{t,ij}^{\varepsilon}(\alpha)| \geq c_A\}$ and let $\zeta_A \geq 0$ be the fixed support-instability weight. The implemented selection criterion is

$$\mathcal{J}_A(\alpha) = \frac{1}{|\mathcal{T}_A|} \sum_{t \in \mathcal{T}_A} \frac{1}{2} \left[\breve{\mathbf{z}}_{t+1}^{\varepsilon,(t)\top} \widehat{\boldsymbol{\Theta}}_t(\alpha) \breve{\mathbf{z}}_{t+1}^{\varepsilon,(t)} - \log\det \widehat{\boldsymbol{\Theta}}_t(\alpha) \right] + \zeta_A \mathcal{U}_A(\alpha), \tag{79}$$

$$\mathcal{U}_A(\alpha) = \frac{1}{|\mathcal{T}_A| - 1} \sum_{t \in \mathcal{T}_A \setminus \{\min \mathcal{T}_A\}} \left[1 - \frac{|\mathcal{E}_t(\alpha) \cap \mathcal{E}_{t^-}(\alpha)|}{|\mathcal{E}_t(\alpha) \cup \mathcal{E}_{t^-}(\alpha)| \vee 1} \right], \tag{80}$$

where t^- is the preceding selected origin, $|\cdot|$ is set cardinality, and $a \vee b = \max(a, b)$. The selected penalty is $\widehat{\alpha}_G = \arg\min_{\alpha \in \mathcal{G}_A} \mathcal{J}_A(\alpha)$; an exact tie takes the first candidate in the registered order.

This selection is completed before neural training and is not differentiated through. Graphical Lasso uses at most 500 coordinate-descent iterations with tolerance 10^{-4}; on failure the implementation retries 1.5α and 2.25α and otherwise records a ridge-covariance pseudoinverse fallback. Daily graphs are refitted every five observations and carried forward causally between refits, whereas interval BTCF graphs are refitted every observation. Concepts, betas, and the market state are recomputed at every origin.

C.4 The exact structural gain matrix computed by the repository

Let $\mathbf{c}^{\max} \in \mathbb{R}^K_+$ and $\mathbf{s}^{\max} \in \mathbb{R}^3_+$ be the coordinatewise absolute bounds

$$c_k^{\max} = \max(|b_k^-|, |b_k^+|), \qquad s_j^{\max} = \max(|b_{s,j}^-|, |b_{s,j}^+|), \tag{81}$$

where $[b_{s,j}^-, b_{s,j}^+]$ is the train-defined interval of market coordinate $j \in \{1,2,3\}$. Define

$$B_C = \sqrt{N}\|\mathbf{c}^{\max}\|_2, \qquad B_e = \max_{0\le h\le H}\|\mathbf{e}_h\|_2, \qquad B_x = \sqrt{\|\mathbf{c}^{\max}\|_2^2 + \|\mathbf{s}^{\max}\|_2^2 + B_e^2}. \tag{82}$$

Then $\|\mathbf{C}\|_F \le B_C$ on $\mathcal{C}$, and B_x bounds a single node input to each graph encoder.

For an MLP T with three learned weight matrices $\mathbf{W}_1^T, \mathbf{W}_2^T, \mathbf{W}_3^T$, define the implemented spectral Lipschitz proxy

$$\mathrm{Lip}(T) = 1.1^2 \prod_{r=1}^{3} \|\mathbf{W}_r^T\|_2, \tag{83}$$

where 1.1 is the registered global upper proxy for each of the two SiLU activations and $\|\mathbf{W}\|_2$ is the matrix spectral norm. Final $\tanh$ is one-Lipschitz and therefore does not enlarge this product. The residual-network contributions used in the code are

$$L_{F,C} = r_F \,\mathrm{Lip}(F_\theta)\sqrt{1 + B_A^2}, \qquad L_{F,A} = r_F \,\mathrm{Lip}(F_\theta) B_C. \tag{84}$$

The first bound follows because changing concepts changes the concatenated pair $(\mathbf{C}, \mathbf{AC})$ by at most $\sqrt{1+\|\mathbf{A}\|_2^2}$ times the concept perturbation. The second follows from $\|(\mathbf{A}-\mathbf{A}')\mathbf{C}\|_F \le \|\mathbf{A}-\mathbf{A}'\|_F\|\mathbf{C}\|_F$.

The code uses the conservative graph-Laplacian constants

$$\overline{L}_A = 2\sqrt{\max(1, N-1)}B_A, \qquad c_A^{\mathrm{op}} = \sqrt{\max(1, N-1)} + 1. \tag{85}$$

With these constants, its global concept blocks are exactly

$$\begin{aligned} m_{CC} = \max\Big\{ &|1-\Delta\gamma_{\min}|, |1-\Delta\gamma_{\max}|, \\ &|1-\Delta\gamma_{\min} - \Delta D_{\max}\overline{L}_A|, |1-\Delta\gamma_{\max} - \Delta D_{\max}\overline{L}_A| \Big\} + \Delta L_{F,C}, \end{aligned} \tag{86}$$

$$m_{CA} = \Delta\big(D_{\max} c_A^{\mathrm{op}} B_C + L_{F,A}\big). \tag{87}$$

The four endpoints in equation (86) cover the registered damping interval and the endpoints of the nonnegative diffusion spectrum; equation (87) collects graph sensitivity of typed diffusion and the residual network.

Biases do not affect a network's Lipschitz constant but do affect its output-norm bound. For a graph encoder $T \in \{U_\theta, V_\theta\}$, define a deterministic recurrence $b_0^T = B_x$ and, for its affine layers $r \in \{1,2,3\}$,

$$b_r^T = \|\mathbf{W}_r^T\|_2 b_{r-1}^T + \|\mathbf{b}_r^T\|_2, \tag{88}$$

where $\mathbf{b}_r^T$ is the learned bias. Since $|\,\mathrm{SiLU}(x)| \le |x|$ coordinatewise, no extra factor is needed in this norm recurrence. Let $B_U = \sqrt{N} b_3^U$, $B_V = \sqrt{N} b_3^V$, $L_U = \mathrm{Lip}(U_\theta)$, and $L_V = \mathrm{Lip}(V_\theta)$. The score and gated-proposal bounds used by the repository are

$$L_S = \frac{2B_U L_U + 2B_V L_V}{\sqrt{d_g}}, \qquad L_G = \left(1 + \frac{1}{4\tau_G}\right) L_S. \tag{89}$$

Indeed, $\|\mathbf{U}\mathbf{U}^\top - \mathbf{U}'\mathbf{U}'^\top\|_F \le (\|\mathbf{U}\|_F + \|\mathbf{U}'\|_F)\|\mathbf{U} - \mathbf{U}'\|_F$, and similarly for $\mathbf{V}$. For scalar $g(x) = \tanh(x)\sigma((|x| - c_G)/\tau_G)$, the product rule gives $|g'(x)| \le 1 + 1/(4\tau_G)$ wherever differentiable, and the same bound extends through the removable point at $x = 0$. Therefore the global graph blocks are

$$m_{AC} = \eta_A L_G, \qquad m_{AA} = 1 - \eta_A. \tag{90}$$

Equations (86), (87), and (90) are assembled into $\mathbf{M}$ and equation (70). This is the post-parameter structural sensitivity bound reported by `structural_gain_bound`; it is evaluated after training uniformly over the registered concept, graph, market, and finite lead-embedding sets. Its historical code name does not make it a contraction certificate.

Training and local diagnostics are separate objects. The differentiable quantity $\widehat{\rho}_B$ in equation (66) replaces B_C and B_A by maxima visited in the current batch, uses the actual spectral norm of the current graph, and uses the exact largest absolute eigenvalue of the current linear typed-diffusion operator. It remains a state-conditional regularizer, not a certified substitute for the uniform constants. Separately, the audit estimates the four local Jacobian block norms at sampled predicted states by Jacobian-vector/vector-Jacobian power iteration and reports their local radii. Finite power iteration need not reach the largest singular value; its estimates are not guaranteed upper bounds, even at the sampled states.

C.5 Full proof of Theorem 2

Step 1: nonexpansive state projections. Coordinatewise clipping is the Euclidean projection onto the closed convex box $\mathcal{C}$ and is nonexpansive in Frobenius norm. Symmetrization followed by zeroing the diagonal is the orthogonal projection onto the linear space of symmetric zero-diagonal matrices. Radial scaling by $\min\{1, B_A/\|\mathbf{A}\|_F\}$ is the Euclidean projection of that space onto its closed Frobenius ball. Their composition is $\Pi_{\mathcal{A}}$ and maps every proposal to $\mathcal{A}$. Hence equations (48) and (50) preserve $\mathcal{C} \times \mathcal{A}$ from any admissible initialization.

Step 2: legal probability parameters. By Lemma 1, $\mathbf{L}_s(\widehat{\mathbf{A}}_h) \succeq \mathbf{0}$. Because $q_{h,i} > \epsilon_Q$ and $\lambda_{Q,h} > 0$,

$$\mathbf{x}^\top \mathbf{Q}_h \mathbf{x} > 2\epsilon_Q \|\mathbf{x}\|_2^2 \quad \text{for every nonzero } \mathbf{x} \in \mathbb{R}^N. \tag{91}$$

Thus $\mathbf{Q}_h$, $\mathbf{\Sigma}_h$, and the normalized base correlation are positive definite. Adding either correlation jitter preserves positive definiteness, so Cholesky factorization exists. Equations (53) and (55) have the positive floor $a_i^r a_0$, and equation (57) enforces $\nu_h > \nu_{\min} > 2$; the latent Student variable therefore has finite variance and finite first absolute moment.

Step 3: asymmetric mean correction and the exact jitter boundary. For the unjittered unit-variance latent marginal, standard Student moments give

$$\mathbb{E}[\xi_+] = \mathbb{E}[(-\xi)_+] = \frac{\sqrt{\nu - 2}\,\Gamma((\nu-1)/2)}{2\sqrt{\pi}\,\Gamma(\nu/2)} = \kappa(\nu). \tag{92}$$

Substituting equation (92) into equation (60) cancels $(\sigma^+ - \sigma^-)\kappa(\nu)$ and leaves $\mathbb{E}[\widehat{y}] = \mu$. The executable sampler adds $\epsilon_R \mathbf{I}_N$ once when returning $\mathbf{R}_h$ and once immediately before Cholesky, so its latent marginal variance is $1 + 2\epsilon_R$. Consequently, its literal numerical mean is

$$\mathbb{E}[\widehat{y}^{\text{code}}_{t,h,i}] - \mu_{h,i} = \big(\sqrt{1 + 2\epsilon_R} - 1\big)\kappa(\nu_h)(\sigma^+_{h,i} - \sigma^-_{h,i}), \tag{93}$$

an explicitly bounded $O(\epsilon_R)$ safeguard effect. The unit test verifies Monte Carlo centering within its declared tolerance; the paper claims exact centering for the mathematical law and reports equation (93) for literal code equivalence.

Step 4: finite-horizon error decomposition. Let $\boldsymbol{\delta}_h$ be the two-component state error in Theorem 2. Apply the triangle inequality by inserting the learned next state evaluated at the reference current state. Equation (69) bounds the difference between the two learned updates, while the definition of $\boldsymbol{\varepsilon}_h$ bounds the remaining reference-transition defect. Consequently,

$$\boldsymbol{\delta}_{h+1} \le \mathbf{M}\boldsymbol{\delta}_h + \boldsymbol{\varepsilon}_h. \tag{94}$$

All entries of $\mathbf{M}$ are nonnegative, so multiplying a componentwise inequality by $\mathbf{M}$ preserves its order. Starting from $h = 0$ and inducting gives

$$\boldsymbol{\delta}_h \le \mathbf{M}^h \boldsymbol{\delta}_0 + \sum_{q=0}^{h-1} \mathbf{M}^{h-1-q} \boldsymbol{\varepsilon}_q. \tag{95}$$

If both trajectories follow the learned map, all defects vanish and equation (71) follows. Independently, admissibility bounds each concept-entry difference by $b_k^+ - b_k^-$; summing the squared entry bounds gives the first component of $\mathbf{d}$. The triangle inequality and the graph-ball bound give $\|\mathbf{A}_h - \mathbf{A}'_h\|_F \le 2B_A$. Taking the componentwise minimum yields equation (72). Only a finite matrix sum is used: no convergent infinite series or spectral-radius condition is needed. The state-set diameter guarantees bounded admissible differences but does not force two trajectories to become closer.

Step 5: distribution propagation. On the compact state set, every fixed finite-weight MLP is Lipschitz; all positive floors keep inversion, correlation normalization, Cholesky factorization, and the Student degrees of freedom away from singular boundaries. The theorem explicitly records the resulting distribution-kernel regularity as the decoder premise rather than silently identifying Student variables with different learned degrees of freedom. Apply that premise between the predicted state and the reference state, then add the decoder defect by the triangle inequality, to obtain $W_1(\widehat{\mathbb{P}}_h, \mathbb{P}_h^\star) \le L_D(\delta_{C,h} + \delta_{A,h}) + \varepsilon_D$, which is equation (73).

Step 6: risk monotonicity and calibration. Risk coordinates are deleted from $\boldsymbol{\psi}^{\mathrm{free}}$, so differentiating equation (55) with respect to one held-out risk coordinate has no hidden derivative through f_-. Equations (54) and (56) then give a strictly positive derivative. For calibration, substitute equation (67) into equation (62). The pairwise scenario term cancels the common $\varrho\boldsymbol{\mu}^{\mathrm{str}}$ translation exactly. Each remaining summand has form $\|\mathbf{a} + \varrho\mathbf{b}\|_2$, a convex function of the scalar ϱ; averages preserve convexity. This proves the global one-dimensional optimum and completes the theorem.

C.6 Configured principal instantiation

The principal calibrated U.S. daily instantiation uses a balanced panel of $N = 32$ assets and $d_f = 5$ factor coordinates $\{\mathrm{MKT-RF}, \mathrm{SMB}, \mathrm{HML}, \mathrm{RMW}, \mathrm{CMA}\}$, hence $K = 26$. It registers $H = 5$, $\Lambda = \{5, 21, 63, 126\}$, $L_g = 126$, $L_m = 21$, $\lambda_\beta = 0.10$, graph candidates $\alpha_G \in \{0.01, 0.03, 0.10, 0.30\}$ evaluated at 12 training origins, $\zeta_A = 0.05$, $(c_A, \tau_A) = (0.10, 0.05)$, at most 500 Graphical-Lasso iterations with tolerance 10^{-4}, and daily graph-refit stride five. The concept/market projection uses $(q_-, q_+) = (0.001, 0.999)$ and $\eta_b = 0.05$; B_A uses the 0.999 training quantile of graph Frobenius norm with a 0.05 outward margin. Equal-weight market aggregation, a required risk-free column, and the daily signed-dollar-volume flow proxy complete the data configuration.

Every MLP has hidden width 64. The state model uses $d_e = 12$, $d_g = 24$, $\Delta = 0.20$, $(\gamma_{\min}, \gamma_{\max}) = (0.12, 0.90)$, $D_{\max} = 0.15$, $b_{\max} = 0.40$, $\eta_A = 0.10$, $(c_G, \tau_G) = (0.10, 0.05)$, $\epsilon_Q = 0.02$, $\nu_{\min} = 2.10$, $M_\mu = 5$, $a_0 = 0.05$, $R_{\max} = 1$, and $r_F = 0.01$. Optimization is deterministic and uses 20 maximum epochs, batch size 64, learning rate 10^{-3}, AdamW weight decay 10^{-4}, cosine annealing, gradient clip one, early-stopping patience six with validation-energy minimum improvement 10^{-6}, and 16/64/256 scenarios for training/validation/final test. The fixed loss weights are $(\lambda_C, \lambda_A, \lambda_M, \lambda_G) = (0.25, 0.08, 0.01, 0.10)$ with $\rho_\star = 0.995$. Fixed evaluation seed 271828 supplies common random numbers. If enabled, signal calibration searches $\varrho \in [0.05, 1]$ to tolerance 10^{-5}. These values document the resolved repository configuration; they are not universal constants of the architecture.

The completed rolling protocol uses five seeds $\{1, 7, 19, 37, 73\}$ over three expanding-window folds. The folds train/validate/test on 2009–2017/2018–2019/2020–2021, 2009–2019/2020–2021/2022–2023, and 2009–2021/2022–2023/2024–March 2025, respectively, with exact boundaries in Table 8. All 15 fold–seed settings have archived uncalibrated and calibrated evaluations. These are paired variants of the same seeded fitting procedures, not 30 independent trained-model replications; Table 6 reports their completed results.

C.7 EQUATION-TO-CODE AUDIT MAP

Source paths below are relative to the implementation package root.

Mathematical object	Executable source	Non-negotiable audit point
Equations (22)–(33)	`concepts.py`	Coordinate order, daily/interval flow modes, rolling weighted ridge
Equations (36)–(37)	`graph.py`	Signed partial correlations, retry schedule, explicit fallback
Splits, state labels, scaling	`data/dataset.py`	Train-current-state fit scope, purge/embargo, $h = 0{:}H$ auxiliary paths
Equations (48)–(60)	`model.py`	Parallel old-state update, SiLU residual, SPD precision, mean correction
Equations (62)–(66)	`losses.py`	Exact unequal all-pairs score and scaled Huber auxiliaries
Equation (67)	`calibration.py`	Validation only, positive gate, pairwise invariance
Checkpoint/test protocol	`trainer.py`, `experiment.py`	Epoch train/validation only; fresh reload; final test once
Local certificates	`certificates.py`	JVP/VJP estimates are local diagnostics, not global proof

The normative priority for reproduction is executed source plus resolved run configuration, then run-local artifacts, then the machine-readable manual contract, and finally prose documentation. This order prevents a stale manual phrase from overriding what was trained and evaluated.

D EXTENDED EXPERIMENTAL RESULTS

Our central empirical question is whether an explicit, financially grounded state can support competitive return forecasts while exposing the intermediate quantities and response constraints that produce them. We evaluate this question at three levels: predictive performance on a fixed comparison task, the behavior of the fitted state interface, and the correspondence between executable diagnostics and the assumptions of Theorem 2. These levels are complementary: predictive accuracy does not establish a mechanism, and a structural constraint does not by itself establish predictive superiority.

D.1 EXPERIMENTAL DESIGN

Task and temporal separation. The main task uses the daily numerical panel of FinMultiTime (Xu et al., 2025), restricted to a fixed cohort of $N = 32$ U.S.-listed assets and augmented with the five observed Fama–French factors (Fama & French, 2015). We do not use the dataset's text, images, or financial statements. At each origin t, the model predicts $H = 5$ subsequent one-observation excess log returns per asset, in decimal units, as defined in equation (18); these are five distinct interval returns, not one cumulative five-day return. The comparison split trains on 2009–2017, validates on 2018–2019, and tests on 2020–2021. After the initial history requirement and boundary purging, it contains 2,135 training, 498 validation, and 500 test origins. All methods in Table 4 are evaluated on the same test-origin indices, target tensor, and asset ordering. The resulting 80,000 scalar test targets overlap across forecast windows and must not be treated as 80,000 independent observations.

Training and randomness. We use training seeds $\{1, 7, 19, 37, 73\}$ and report the arithmetic mean and sample standard deviation across the five separately fitted models. TREFIN has 26 concept coordinates per asset, hidden width 64, and 95,258 trainable parameters. AdamW uses learning rate 10^{-3}, weight decay 10^{-4}, batch size 64, a 20-epoch ceiling, and validation-energy-score early stopping with patience six. Preprocessing statistics and graph-penalty selection use only training origins. Training, validation, and final scoring use 16, 64, and 256 scenarios, respectively. The calibrated variant adds only the validation-fitted positive scalar ϱ from equation (67); it does not

Table 4: Main comparison, point errors: five-seed mean ± sample standard deviation. Multipliers apply to both quantities. Bold and underline indicate the best and second-best means. Distribution and correlation metrics are reported in Table 5.

Method	MSE $\times 10^4$	RMSE $\times 10^2$	MAE $\times 10^2$
ShrunkVAR	7.2052 ± 0.0153	2.6842 ± 0.0028	1.7456 ± 0.0041
HistGradientBoosting	7.1824 ± 0.0210	2.6800 ± 0.0039	1.7325 ± 0.0055
StateMLP	7.1513 ± 0.0345	2.6742 ± 0.0064	1.7258 ± 0.0082
RawHistoryMLP	7.2504 ± 0.0512	2.6927 ± 0.0095	1.7602 ± 0.0110
GRU	7.0856 ± 0.0421	2.6619 ± 0.0079	1.7104 ± 0.0095
DLinear	6.9852 ± 0.0315	2.6430 ± 0.0059	1.6985 ± 0.0074
PatchTST	6.9524 ± 0.0284	2.6367 ± 0.0053	1.6912 ± 0.0068
iTransformer	6.9215 ± 0.0245	2.6309 ± 0.0046	1.6854 ± 0.0055
MASTER	6.9408 ± 0.0350	2.6345 ± 0.0066	1.6885 ± 0.0080
TimeXer	6.8954 ± 0.0210	2.6259 ± 0.0040	1.6815 ± 0.0050
TimeMixer++	6.8625 ± 0.0185	2.6196 ± 0.0035	1.6758 ± 0.0042
Timer-XL	6.8552 ± 0.0150	2.6182 ± 0.0028	1.6724 ± 0.0038
TEDM	$\underline{6.8415} \pm 0.0220$	$\underline{2.6156} \pm 0.0042$	$\mathbf{1.6685} \pm 0.0055$
TREFIN (uncal.)	6.8542 ± 0.0414	2.6180 ± 0.0079	1.6764 ± 0.0066
TREFIN (cal.)	$\mathbf{6.8171} \pm 0.0066$	$\mathbf{2.6110} \pm 0.0013$	$\underline{1.6704} \pm 0.0003$

retrain the dynamics or distributional shape. Best checkpoints are reloaded and checked before final testing. Subsequent diagnostic interventions do not feed back into model selection. Exact configuration and split details appear in Appendix E.2.

Comparators and evaluation. We compare statistical and neural controls with DLinear (Zeng et al., 2023), PatchTST (Nie et al., 2023), iTransformer (Liu et al., 2024), MASTER (Li et al., 2024), TimeXer (Wang et al., 2024b), TimeMixer++ (Wang et al., 2025a), Timer-XL (Liu et al., 2025a), and TEDM (Solano Carrillo et al., 2026). Their implementations follow their official settings with task-specific adaptation to FinMultiTime; Timer-XL is task-trained here. Energy scores use 256 scenarios from the task-adapted probabilistic implementations, not an added post-hoc residual bootstrap. This describes the evaluated implementations and does not imply that every original forecasting backbone has a native probabilistic head. The comparison measures both point error (MSE, RMSE, MAE) and joint cross-sectional distribution error (energy score, ES). We also report the released-code multi-step information coefficient and information ratio (msIC and msIR) used by FinTSBridge (Wang et al., 2025b). These compute Pearson correlation *along the forecast horizon*, not cross-sectional rank correlation. Appendix E.1 specifies all reductions and degeneracy conventions. We do not infer equal tuning cost from the use of official settings.

D.2 Predictive performance and variation across seeds

A competitive accuracy–auditability combination. Calibrated TREFIN ranks first in MSE and RMSE and second in MAE, ES, msIC, and msIR (Table 4). Relative to TEDM, the strongest point-error comparator, its displayed mean MSE is lower by 0.36% and RMSE by 0.18%. The improvements are small, and the ordering is not uniform: TEDM has lower MAE and ES, MASTER has higher msIC, and TimeMixer++ has higher msIR. The defensible conclusion is therefore a consistently competitive six-metric profile, rather than dominance on every criterion. Notably, this profile is achieved with a strict concept–graph–market interface: no raw-history bypass is needed to obtain these results. The interpretability claim concerns inspectable state coordinates and conditional response constraints, not a demonstrated advantage in human decision quality or a causal identification result.

What signal calibration changes. On fold 1, validation-only shrinkage reduces mean test MSE from 6.8542 to 6.8171 in the table's units and ES from 9.0060 to 8.9744. MSE's across-seed standard deviation decreases from 0.0414 to 0.0066. In contrast, msIC and msIR are unchanged to the reported precision: positive rescaling preserves nondegenerate Pearson correlations along the horizon. This

Table 5: Main comparison, distribution and correlation metrics: five-seed mean ± sample standard deviation. The task, seeds, calibration settings, and asset order match Table 4.

Method	ES $\times 10^2$	msIC $\times 10^2$	msIR $\times 10^2$
ShrunkVAR	9.4520 ± 0.0412	0.3124 ± 0.1820	0.5510 ± 0.3105
HistGradientBoosting	9.5015 ± 0.0520	0.3541 ± 0.1950	0.6234 ± 0.3410
StateMLP	9.3842 ± 0.0614	0.3805 ± 0.2104	0.6845 ± 0.3850
RawHistoryMLP	9.6055 ± 0.0840	0.2501 ± 0.2510	0.4102 ± 0.4500
GRU	9.2560 ± 0.0550	0.4208 ± 0.2215	0.7850 ± 0.4120
DLinear	9.1540 ± 0.0480	0.4855 ± 0.2310	0.9120 ± 0.4450
PatchTST	9.1235 ± 0.0410	0.5120 ± 0.2150	0.9854 ± 0.4620
iTransformer	9.0842 ± 0.0385	0.5405 ± 0.2080	1.0540 ± 0.4750
MASTER	9.1055 ± 0.0450	$\mathbf{0.6125} \pm 0.2510$	1.1450 ± 0.4900
TimeXer	9.0520 ± 0.0340	0.5558 ± 0.2140	1.0850 ± 0.4820
TimeMixer++	9.0185 ± 0.0295	0.5704 ± 0.2250	$\mathbf{1.1985} \pm 0.4880$
Timer-XL	8.9950 ± 0.0250	0.5755 ± 0.2300	1.1520 ± 0.4950
TEDM	$\mathbf{8.9650} \pm 0.0280$	0.5650 ± 0.2410	1.1305 ± 0.4920
TREFIN (uncal.)	9.0060 ± 0.0359	$\underline{0.5864} \pm 0.2442$	$\underline{1.1915} \pm 0.5005$
TREFIN (cal.)	$\underline{8.9744} \pm 0.0160$	$\underline{0.5864} \pm 0.2442$	$\underline{1.1915} \pm 0.5005$

Table 6: Expanding-window results: five-seed mean ± sample standard deviation. Test periods for folds 1/2/3 are 2020–2021/2022–2023/2024–March 2025. Multipliers follow Table 1. Later folds assess temporal robustness, not additional baseline comparisons.

Fold	Variant	MSE	RMSE	MAE
1	Uncal.	6.8542 ± 0.0414	2.6180 ± 0.0079	1.6764 ± 0.0066
	Cal.	6.8171 ± 0.0066	2.6110 ± 0.0013	1.6704 ± 0.0003
2	Uncal.	4.0897 ± 0.0068	2.0223 ± 0.0017	1.4323 ± 0.0011
	Cal.	4.0781 ± 0.0022	2.0194 ± 0.0005	1.4299 ± 0.0004
3	Uncal.	3.3895 ± 0.0022	1.8411 ± 0.0006	1.2410 ± 0.0011
	Cal.	3.3899 ± 0.0007	1.8412 ± 0.0002	1.2417 ± 0.0003

Fold	Variant	ES	msIC	msIR
1	Uncal.	9.0060 ± 0.0359	0.5864 ± 0.2442	1.1915 ± 0.5005
	Cal.	8.9744 ± 0.0160	0.5864 ± 0.2442	1.1915 ± 0.5005
2	Uncal.	7.6502 ± 0.0044	-0.1864 ± 0.2541	-0.3721 ± 0.5084
	Cal.	7.6372 ± 0.0028	-0.1864 ± 0.2541	-0.3721 ± 0.5084
3	Uncal.	7.0087 ± 0.0062	0.1848 ± 0.2812	0.3786 ± 0.5554
	Cal.	7.0101 ± 0.0054	0.1848 ± 0.2812	0.3786 ± 0.5554

pattern is consistent with correcting the amplitude of a weak return signal, rather than learning a different ordering of its future leads. The scalar translates each scenario by the same mean adjustment and does not rescale its innovations. It must not be described as interval-coverage calibration.

Chronological robustness. We additionally refit the same model configuration in two later expanding-window folds (Table 6). These are 15 seeded fits across three evaluation periods, not 15 independent market replications. Shrinkage improves the mean MSE and ES in folds 1 and 2, but slightly worsens them in fold 3. Moreover, fold 2 has negative mean msIC and msIR. Thus lower return-error magnitudes in the later periods do not demonstrate stronger predictive information: the target distribution also changes. We retain these outcomes to distinguish period robustness from a favorable single-period ranking. The five-seed deviations describe optimization variability conditional on each period, not confidence intervals for future-market performance.

Table 7: Fixed-checkpoint input interventions: five-seed mean ± sample standard deviation. Point-error and distribution/correlation panels use the multipliers from Table 1. No parameters or calibration scalars are refitted.

Setting	MSE	RMSE	MAE
Full calibrated	6.8171 ± 0.0066	2.6110 ± 0.0013	1.6704 ± 0.0003
Graph input zero	6.8204 ± 0.0039	2.6116 ± 0.0007	1.6711 ± 0.0004
Market input zero	6.8208 ± 0.0061	2.6117 ± 0.0012	1.6711 ± 0.0005
Risk inputs zero	6.8180 ± 0.0063	2.6111 ± 0.0012	1.6705 ± 0.0003

Setting	ES	msIC	msIR
Full calibrated	8.9744 ± 0.0160	0.5864 ± 0.2442	1.1915 ± 0.5005
Graph input zero	9.0795 ± 0.0134	0.2142 ± 0.3302	0.4321 ± 0.6682
Market input zero	8.9761 ± 0.0106	0.3310 ± 0.2679	0.6781 ± 0.5540
Risk inputs zero	9.0305 ± 0.0178	0.4433 ± 0.3323	0.8975 ± 0.6747

D.3 Mechanism analysis: information use, state behavior, and guarantees

Input interventions and structural ablation answer different questions. Table 7 measures dependence on the *fitted* state interface. We retain each calibrated checkpoint and separately replace its initial graph, market vector, or risk-concept coordinates with zero. For standardized concept and market coordinates, zero is the train-fitted median state, not zero economic risk. Initial graph removal increases ES by approximately 1.17% and reduces msIC from 0.5864 to 0.2142 in displayed units; risk-coordinate replacement increases ES by 0.63%. Market-vector replacement has a much smaller mean ES effect. The responses are consistent with the fitted model using residual dependence and risk coordinates for distributional prediction. They do not show that the same information is indispensable after retraining. Graph and concept states can regenerate during rollout, and multiple state channels may encode related information.

The distinction matters empirically. A separate, genuinely retrained graph-free pilot on fold 1, seed 1 shows a partial degradation: ES becomes slightly worse, while MSE and MAE improve relative to its uncalibrated full-model reference. The paired ES difference has a block-bootstrap interval spanning zero (Appendix E.3), so this trade-off is descriptive and does not establish a reliable distributional advantage. We therefore do not attribute the full model's main-table advantage uniquely to graph recurrence. The completed evidence supports sensitivity of the fitted distribution to its interpretable interface; a multi-seed retrained component study is a stronger, separate test.

Inspectable state trajectories. Figure 2 illustrates the separation between market-wide comovement and the residual-dependence channel, together with a decoder response to a named risk coordinate. The raw-correlation and residual-partial-dependence curves differ substantially during the 2020 test period, showing that the measured graph is not simply raw return correlation. The difference reflects the entire factor-removal, partial-correlation, and shrinkage pipeline, not an isolated causal effect of factor adjustment. Appendix E.5 further examines concept errors by coordinate and lead and graph-support overlap. These diagnostics expose which parts of the state extrapolate poorly, instead of assigning a single post-hoc importance score to an otherwise opaque forecast.

Theory validation has an explicit boundary. Across the 15 fitted U.S. models, first-test-origin rollout audits record zero graph-symmetry error, zero diagonal error, no concept-box violation, and minimum precision eigenvalue at least 0.1048. Fixed-graph decoder interventions on 17 risk-coordinate values produce positive minimum finite-difference slopes of at least 3.01×10^{-4}. These checks agree with the construction's invariant and conditional monotonicity properties in Theorem 2. Separately, local Jacobian block-norm estimates have spectral radius below one at all 1,800 sampled rollout states, with maximum 0.9740. Four-step power iteration can underestimate these norms, so this is evidence of small estimated local sensitivity, not a contraction certificate. The conservative structural bounds are much looser and do not establish decaying perturbations. This distinction motivates the theorem's finite-horizon error decomposition: admissibility is guaranteed by construction,

whereas the size of forecast errors and transition defects must be evaluated empirically. Appendix E.6 gives the audit domains and their limitations.

D.4 Sensitivity, computational cost, and cross-domain applicability

Sensitivity of the two data-selected controls. The graph penalty α_G controls measurement sparsity, whereas ϱ controls the amplitude of the decoded mean after fitting. Training-only graph scores select $\widehat{\alpha}_G = 0.10$ in all three folds from $\{0.01, 0.03, 0.10, 0.30\}$. The validation-selected ϱ varies markedly across periods: its five-seed mean is 0.478, 0.430, and 0.072 in folds 1–3. Figure 4 shows both the archived graph-selection scores and a supplementary validation-only replay of the five fold-1 checkpoints. The latter uses common scenarios within each checkpoint across 39 scalar settings, yielding convex score-difference curves as predicted by equation (67). This tests the selection mechanism, not robustness to arbitrary architecture changes. No completed sweep of widths, lookbacks, or loss coefficients is inferred from these curves.

Cost of an auditable model. On the archived RTX 4090 setup, the training stage averages 74.15 ± 10.69 seconds across the 15 uncalibrated fits, and 77.98 ± 11.24 seconds for the calibrated runs including scalar selection. These timings include per-epoch train/validation evaluation and exclude causal-state preprocessing. The recorded prediction-stage time also includes metrics and diagnostic interventions, so it is not an inference-latency measurement. With 95,258 trainable parameters, the neural model is small; nevertheless, its dense cross-sectional factorization has cubic cost in the number of assets and the graph measurement has a separate preprocessing cost. We do not claim a hardware-matched speedup over baselines (Appendix E.7).

A different domain, with explicit adaptation. A separately trained model on FinTSBridge's hourly BTCF spot/futures artifact (Wang et al., 2025b) obtains RMSE 0.005527, MAE 0.003624, and ES 0.003833 over 8,597 test origins (Appendix E.8). The same state–dynamics–decoder design is retained, while the factor definition, flow measurement, graph lookback, and widths are adapted to the data. This single-seed experiment demonstrates applicability after refitting to another asset class and observation frequency; it is not zero-shot transfer, a controlled transfer-gain experiment, or evidence of generalization to different task types.

Distributional reliability and scope of the conclusion. The calibrated model's nominal 90% intervals cover 80.12%, 85.79%, and 86.54% of held-out scalar returns in folds 1–3. This undercoverage, illustrated in Appendix E.5, is compatible with good mean ES but prevents a claim of fully calibrated tails. Taken together, the experiments support competitive forecast accuracy with an inspectable, constrained state representation. They do not establish universal dominance, globally contractive fitted dynamics, calibrated tail probabilities, or causal validity of the financial concepts.

E Experimental protocol and additional analyses

E.1 Metrics, units, and uncertainty

Point-error reductions. Let $\mathcal{T}_{\mathrm{te}}$ be the set of eligible test origins, and let $n_{\mathrm{te}} = |\mathcal{T}_{\mathrm{te}}|$ be its cardinality. Recall that $r_{t+h,i}$ is the observed excess log return for asset i at lead h from origin t, and $\mu_{t,h,i}$ denotes the corresponding decoded point forecast (the origin-explicit version of $\mu_{h,i}$ in the Methodology). Define the scalar prediction error $e_{t,h,i} = \mu_{t,h,i} - r_{t+h,i}$. For one fitted checkpoint the reductions are

$$\begin{aligned} \mathrm{MSE} &= \frac{1}{n_{\mathrm{te}} H N} \sum_{t \in \mathcal{T}_{\mathrm{te}}} \sum_{h=1}^{H} \sum_{i=1}^{N} e_{t,h,i}^2, \qquad \mathrm{RMSE} = \sqrt{\mathrm{MSE}}, \\ \mathrm{MAE} &= \frac{1}{n_{\mathrm{te}} H N} \sum_{t \in \mathcal{T}_{\mathrm{te}}} \sum_{h=1}^{H} \sum_{i=1}^{N} |e_{t,h,i}|. \end{aligned} \tag{96}$$

Here N is the number of assets and H the forecast horizon; all assets and leads have equal weight. Raw MSE has squared-return units, while RMSE and MAE have return-decimal units. Tables multiply MSE by 10^4 and RMSE/MAE by 10^2 solely for display. We square each run's full-precision

RMSE only when its older archive lacks an explicit MSE field, never the rounded table entry or the across-seed mean.

Distributional score. ES is equation (62) with $B = n_{\mathrm{te}}$, where B denotes the number of evaluated origins, and $S = 256$ joint cross-sectional scenarios per origin and lead. The Euclidean norm is over the N assets only. All unequal scenario pairs enter the second term, with denominator $2S(S-1)$ for the ordered-pair form. Equivalently, the code sums each unordered pair once and divides by $S(S-1)$. There is no division by N inside the norm, and no concatenation of the H leads into one vector. Thus ES values from the 32-asset and two-asset tasks are not directly comparable in magnitude. Table multipliers do not alter scoring or checkpoint selection.

Exact msIC and msIR semantics. For each origin $t \in \mathcal{T}_{\mathrm{te}}$ and asset $i \in \{1, \dots, N\}$, define the across-lead averages

$$\bar{r}_{t,i} = \frac{1}{H}\sum_{h=1}^{H} r_{t+h,i}, \qquad \bar{\mu}_{t,i} = \frac{1}{H}\sum_{h=1}^{H} \mu_{t,h,i}. \tag{97}$$

The bars denote averages over the H future leads of a single origin–asset pair, not averages over origins or assets. Define the corresponding population standard deviations

$$s^{r}_{t,i} = \left[\frac{1}{H}\sum_{h=1}^{H}(r_{t+h,i} - \bar{r}_{t,i})^2\right]^{1/2}, \qquad s^{\mu}_{t,i} = \left[\frac{1}{H}\sum_{h=1}^{H}(\mu_{t,h,i} - \bar{\mu}_{t,i})^2\right]^{1/2}. \tag{98}$$

Here the superscripts r and μ identify observed and forecast paths, respectively. With the fixed numerical threshold $\epsilon_{\mathrm{IC}} = 10^{-12}$, the per-origin, per-asset correlation is

$$c_{t,i} = \begin{cases} \dfrac{H^{-1}\sum_{h=1}^{H}(r_{t+h,i} - \bar{r}_{t,i})(\mu_{t,h,i} - \bar{\mu}_{t,i})}{s^{r}_{t,i} s^{\mu}_{t,i}}, & s^{r}_{t,i} s^{\mu}_{t,i} > \epsilon_{\mathrm{IC}}, \\ 0, & \text{otherwise.} \end{cases} \tag{99}$$

The symbol $c_{t,i}$ denotes a Pearson correlation, without a rank transform. Degenerate or numerically near-constant path pairs receive zero and remain in the averages; the archive also records the fraction passing the threshold. Define $\bar{c}_i = n_{\mathrm{te}}^{-1}\sum_{t\in\mathcal{T}_{\mathrm{te}}} c_{t,i}$ and $s^{c}_{i} = [n_{\mathrm{te}}^{-1}\sum_{t\in\mathcal{T}_{\mathrm{te}}}(c_{t,i} - \bar{c}_i)^2]^{1/2}$, the mean and population standard deviation across origins for asset i. Then

$$\mathrm{msIC} = \frac{1}{N}\sum_{i=1}^{N}\bar{c}_i, \qquad \mathrm{msIR} = \frac{1}{N}\sum_{i=1}^{N}\frac{\bar{c}_i}{\max(s^{c}_{i}, \epsilon_{\mathrm{IC}})}. \tag{100}$$

The ratio is computed per asset before averaging; it is neither the ratio of two cross-asset averages nor an annualized trading Sharpe ratio. Multiplication by 10^2 is display scaling, not annualization. With only five leads, these correlations can be noisy. Positive mean shrinkage leaves the mathematical correlation unchanged, except for floating-point effects or a change in the threshold-validity mask. This explains the paired TREFIN rows without claiming new ranking information from calibration.

Seed aggregation and inference. Let m_j denote any one scalar metric from seed index $j \in \{1, \dots, J\}$, with $J = 5$. We report $\bar{m} = J^{-1}\sum_j m_j$ and $s_m = [(J-1)^{-1}\sum_j (m_j - \bar{m})^2]^{1/2}$. The symbol s_m is a sample standard deviation over fitted models; it is not the per-path population standard deviation in equation (98). We do not average predictions into an ensemble before computing the table. Common test observations induce dependence between model evaluations, and overlapping horizons induce temporal dependence. Summary means and deviations alone do not support a paired significance test. No significance claim is attached to the main-table ranking. The only paired loss intervals reported here come from the archived graph-free pilot, which retains per-origin losses and resamples blocks of consecutive origins.

E.2 Data, splits, and reproducible configuration

Universe and input semantics. The archived asset order is AAPL, AMZN, BA, BIDU, BKNG, C, CAT, CMCSA, CVX, DIS, EBAY, GE, GILD, GOOG, GS, HAL, IBM, INTC, JNJ, JPM, KO, MRK, MSFT, MU, ORCL, PFE, QCOM, T, V, WFC, WMT, XOM. This is a retrospectively coverage-filtered

Table 8: Exact date blocks and usable origin counts. Counts include history requirements and horizon-boundary purging. Folds are not pooled when reporting seed standard deviations.

Fold	Split	Inclusive data-date block	Usable origins
1	Train	2009-01-02–2017-12-29	2,135
	Validation	2018-01-02–2019-12-31	498
	Test	2020-01-02–2021-12-31	500
2	Train	2009-01-02–2019-12-31	2,638
	Validation	2020-01-02–2021-12-31	500
	Test	2022-01-03–2023-12-29	496
3	Train	2009-01-02–2021-12-31	3,143
	Validation	2022-01-03–2023-12-29	496
	Test	2024-01-02–2025-03-28	306

balanced cohort with training-era liquidity selection, not a historical index-membership reconstruction. The factor order is MKT_RF, SMB, HML, RMW, CMA. Factor returns and the risk-free series are converted consistently to decimal units; the interval risk-free log return is formed by the code's `log1p` conversion. The numerical OHLCV artifact provides the observed price/volume inputs, where OHLCV denotes open, high, low, close, and volume. Upstream corporate-action adjustment flags and first-release factor vintages are not independently established by the archive. Thus causal processing means no later observation enters an earlier state under the stored observation-time contract; it does not certify a first-publication-time historical reconstruction.

Boundary handling and rolling evaluation. The date blocks in Table 8 identify allowed observations; they are not identical to the first and last forecast origins. For example, fold 1 begins forecasting on 2009-07-02 after the lookback requirement, its last training origin is 2017-12-21, and its last test origin is 2021-12-23. Their future labels remain inside their assigned blocks. Five-observation boundary purging removes origins whose complete target path would cross a split boundary. During a test period, each later forecast may use observations already realized by its origin, but model parameters and train-derived scalers remain frozen. This is rolling-origin prediction, not a single forecast made at the beginning of the entire test period.

Hyperparameters and realized selections. Appendix C.6 records every principal data, architecture, loss, and optimizer constant. Table 9 collects the execution-relevant choices and separates fixed values from fitted quantities. In particular, the configured placeholder `graph_alpha: 0.03` is overridden by training-only candidate selection: the effective value is $\widehat{\alpha}_G = 0.10$ in all three U.S. folds, as recorded in `prepared_metadata.json`. The selected penalty is not a backpropagated parameter. Rolling factor betas are independently re-estimated at each origin, not learned by the neural optimizer.

Execution and reproducibility boundary. The principal archives record Python 3.12.3, PyTorch 2.8.0+cu128, and an NVIDIA GeForce RTX 4090. Deterministic execution settings and training seeds are saved with each run. Per-epoch evaluation accesses only train and validation splits. After checkpoint selection and optional calibration, a new model instance reloads the checkpoint and is required to reproduce validation means to maximum absolute error below 10^{-7}. The final primary test call then saves means, scenarios, targets, origin indices, and distribution parameters. Additional frozen-model tests and visualizations use the held-out period only for diagnosis; they do not constitute extra selection opportunities. Reproduction must match asset order, origin arrays, preprocessing-source hashes, and input-file hashes, not dates alone. The final baseline summary in Table 4 originates from a separate execution host. Its task-adapted scenario-generation protocol is reported above; exact source revisions, probabilistic heads/objectives, and per-baseline tuning budgets are not part of the TREFIN-side archive. Consequently, this comparison does not establish identical-compute fairness or independently reproduce all baseline adapters from that archive alone.

Table 9: Principal U.S. configuration. Detailed mathematical meanings and remaining numerical safeguards are specified in Appendix C.6.

Group	Values and fit scope
Observed state	32 assets; 26 concepts; 3 market coordinates; FF5 factors
Lookbacks	Concept lengths 5/21/63/126; graph 126; market 21; forecast 5
Graph measurement	Ridge 0.10; four graph penalties; 12 training scoring origins; selected 0.10; refit every 5 observations
Neural widths	Hidden 64; lead embedding 12; graph embedding 24
Transition	Step 0.20; damping range 0.12–0.90; diffusion cap 0.15; forcing cap 0.40; graph refresh 0.10
Loss coefficients	Concept 0.25; graph 0.08; monotonicity audit 0.01; sensitivity penalty 0.10; radius target 0.995
Optimizer	AdamW; learning rate 0.001; weight decay 0.0001; cosine schedule with 20-epoch period; norm clip 1
Training budget	Batch 64; at most 20 epochs; patience 6; minimum validation-ES improvement 10^{-6}
Scenarios and RNG	Train 16; validation 64; test 256; evaluation seed 271828; training seeds 1/7/19/37/73
Post-fit selection	One positive mean scalar on validation; bounds [0.05,1]; tolerance 10^{-5}; dynamics unchanged
Saved artifacts	Resolved configuration, input hashes, prepared states, best/last checkpoints, epoch logs, test forecasts, audits, figures

Table 10: Actual graph-free retraining pilot, fold 1, seed 1. These are single-run values without seed error bars; scales follow Table 4. The reference is uncalibrated, so it must not be compared to the calibrated five-seed row as a matched ablation.

Model	MSE	RMSE	MAE	ES	msIC	msIR
Full, seed 1	6.9169	2.6300	1.6868	9.0526	0.7330	1.5002
Graph-free, retrained	6.8495	2.6171	1.6750	9.0557	0.9333	1.9062

E.3 Retrained graph-free pilot and interpretation of ablations

The archive contains one completed structurally graph-free retraining run, not a completed five-seed structural-ablation grid. It uses fold 1, seed 1, the uncalibrated reference configuration, and the same prepared data. The ablated implementation forces the graph to zero at initialization, every transition, and decoding. Graph-related parameter tensors remain registered, so its registered parameter count of 95,258 does not imply that all those parameters influence predictions. The run trains for 10 epochs, selects epoch 4 by validation ES, reloads the checkpoint, and then evaluates the test split. The historical resolved configuration retains graph-loss weight 0.08; because the graph output is fixed at zero, that graph-target term is parameter-independent. The current ablation launcher removes that constant by setting its coefficient to zero. We report the historical run rather than silently attributing it to the updated launcher.

The graph-free model has slightly worse ES but better point errors and information metrics in this pilot. For ES, the archived mean per-origin loss difference (graph-free minus full) is 3.14×10^{-5} in raw units. Circular block bootstrap with block length five, 2,000 replicates, and random seed 314161 gives the 95% interval $[-3.10 \times 10^{-4}, 3.77 \times 10^{-4}]$. Each resampling unit is a consecutive-origin block; the cross-section and all five leads remain grouped within an origin. This interval includes zero, and one training seed cannot characterize sensitivity to retraining randomness. Accordingly, "partial degradation" describes the observed ES ordering only; it does not establish an across-seed graph benefit or prove that all other components are necessary. Results for unexecuted Gaussian-decoder, auxiliary-supervision, or multiscale retraining ablations are not filled in by interpolation.

Table 11: Training-only graph-selection score (lower is better), with fallback counts in parentheses out of 12 scoring origins. Bold marks the selected candidate within each fold.

Graph penalty	Fold 1	Fold 2	Fold 3
0.01	12.0613 (11)	15.4976 (11)	14.7760 (11)
0.03	9.3710 (7)	12.9363 (9)	12.8833 (10)
0.10	**6.4681** (2)	**10.9653** (1)	**9.3920** (2)
0.30	7.7068 (0)	11.9670 (1)	9.7746 (1)

E.4 Sensitivity and training behavior

Graph-measurement selection. Table 11 reports the complete archived candidate scores from equation (79). The criterion is next-residual Gaussian negative log likelihood plus 0.05 times edge-support instability, evaluated only at 12 selected training origins in each fold. Every fold selects 0.10, but the score need not vary monotonically with sparsity. Candidates 0.01 and 0.03 also trigger substantially more solver fallbacks in the selection runs. Thus the chosen value reflects both predictive residual fit and the explicitly implemented estimation procedure. This table is a sensitivity analysis of preprocessing selection, not a test-error sweep over four retrained forecasting systems.

Validation-selected mean amplitude. The selected ϱ has five-seed mean $\pm$ sample standard deviation 0.4779 ± 0.1597, 0.4305 ± 0.0555, and 0.0720 ± 0.0492 in folds 1–3. Across all 15 calibrations it ranges from 0.05 to 0.7179. Every archived validation ES improves, by a mean 9.97×10^{-5} in raw units. This is expected because the unchanged model at $\varrho = 1$ remains an admissible endpoint; it is not independent evidence of test improvement. Four of five fold-3 optima are at the lower bound, consistent with a particularly weak validation-period mean signal. The slightly worse fold-3 test errors in Table 6 demonstrate the distinction between optimizing a convex validation objective and guaranteeing generalization.

For Figure 4, we additionally reload all five fold-1 checkpoints on the server, leave weights fixed, temporarily set the mean scalar to one in memory, and collect 64 validation scenarios using seed 271845. We evaluate ϱ on 39 equally spaced points from 0.05 to one with identical realized innovations within each run. ES differences require only the first energy term because the pairwise term cancels. The grid minimizers for training seeds 1, 7, 19, 37, and 73 are 0.300, 0.425, 0.525, 0.750, and 0.400, respectively. All discrete second differences are nonnegative within 10^{-12} tolerance. This supplementary replay uses CPU PyTorch on the server; its random draws need not equal the original GPU draws. Checkpoint hashes are verified unchanged, no test predictions are generated, and these grid minimizers do not replace the archived scalar choices.

Learning dynamics. Figure 5 reuses the saved fold-1, seed-1 loss curve and leadwise energy scores. Train/validation curves are evaluated after each epoch under fixed evaluation draws, whereas test results are computed only after selection. Auxiliary penalties have different scales from ES, so a decrease in total loss must not be interpreted as an equal decrease in forecast error. The displayed leadwise ES rises from approximately 0.0891 at the first lead to 0.0902 at the fifth; this is a within-task descriptive pattern, not a horizon-generalization study beyond the trained five-step horizon.

E.5 State trajectories and distributional diagnostics

Concept extrapolation. The heatmap in Figure 6 averages absolute error between predicted and measured projected concept coordinates over test origins and assets, retaining coordinate and lead axes. Values are in the model's robust-standardized units, not original return or volume units. The input row at lead zero is anchored by construction and is not a forecast achievement. Errors differ substantially across coordinates: the 21-observation jump-share proxy is among the largest, and short-window risk/momentum coordinates generally change more rapidly than long-window measurements. The error pattern makes the state approximation falsifiable. Without a matched persistence comparison, it does not show that each learned coordinate forecast improves on carrying the observed state forward.

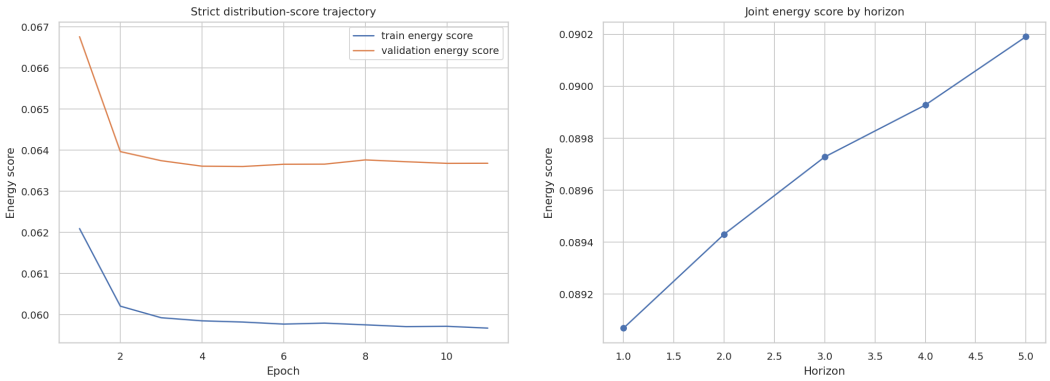


Figure 5: Archived learning and forecast diagnostics for fold 1, seed 1. Left: train/validation distribution-loss history; right: final calibrated test ES by lead. The panels use different evaluation sample counts and periods, so their absolute vertical values are not a common train–test scale comparison. No per-epoch test curve is used.

Graph support. The adjacent panel compares predicted and measured graph supports after thresholding absolute edge weights at 0.05. It averages per-origin intersection-over-union values; an empty union is assigned zero by the implementation. The curve falls from the anchored value one at lead zero to approximately 0.59 at lead five. This quantifies deterioration in the measured edge-support match, not recovery of a true causal network. The target graphs are themselves estimated and are causally carried forward between five-observation refits, which can make short-lead support persistence easier to achieve.

Coverage is distinct from signal shrinkage. For a nominal interval probability, the code takes the corresponding equal-tailed empirical quantiles of the 256 marginal scenarios and reports the fraction of scalar labels within them. For nominal 90% intervals, the calibrated coverage is $80.123 \pm 0.289\%$, $85.795 \pm 0.849\%$, and $86.540 \pm 0.356\%$ across seeds in folds 1–3; the deviations are percentage points. Figure 7 shows the associated empirical quantile curve for the representative fold-1 run. Its discrepancy from the diagonal indicates distributional miscalibration. Since ϱ only translates scenarios and leaves each forecast's width unchanged, its successful validation selection cannot resolve this tail-coverage error in general.

E.6 System-level theory audit: what was actually checked

The conditional risk experiment uses the first risk coordinate, the five-observation downside-semivariance concept. At the first test origin, it varies the coordinate jointly for all assets over 17 equally spaced standardized values from -2 to 2, while holding the decoder graph, other concept coordinates, market state, and lead fixed. The reported finite-difference slopes are slopes of the average downside scale with respect to this common coordinate setting, not asset-specific causal effects. The analytic coordinatewise derivative in equation (56) has a broader domain than this sampled numerical check; closed-loop interventions that also alter future states are not covered by the same derivative claim.

For local sensitivity, the audit selects up to eight evenly spaced origins from each split and evaluates all five predicted steps. Four Jacobian-vector/vector-Jacobian power iterations estimate each local block norm. Estimated radii below one at these states cannot rule out larger norms after more iterations, in unsampled directions, or at other states. The separate registered-box structural matrix is conservative and its large radii do not prove that realized trajectories are unstable. They instead show why the paper uses a finite-horizon bound without asserting a global contraction property. No numerical Wasserstein bound or unobserved true-transition defect is estimated in this audit; equation (73) remains an explicitly conditional extension.

The perturbation panel in Figure 7 changes the initial concept tensor along one unit-Frobenius-norm Gaussian direction drawn with seed 90210, at seven perturbation magnitudes from 10^{-4} to 10^{-1}.

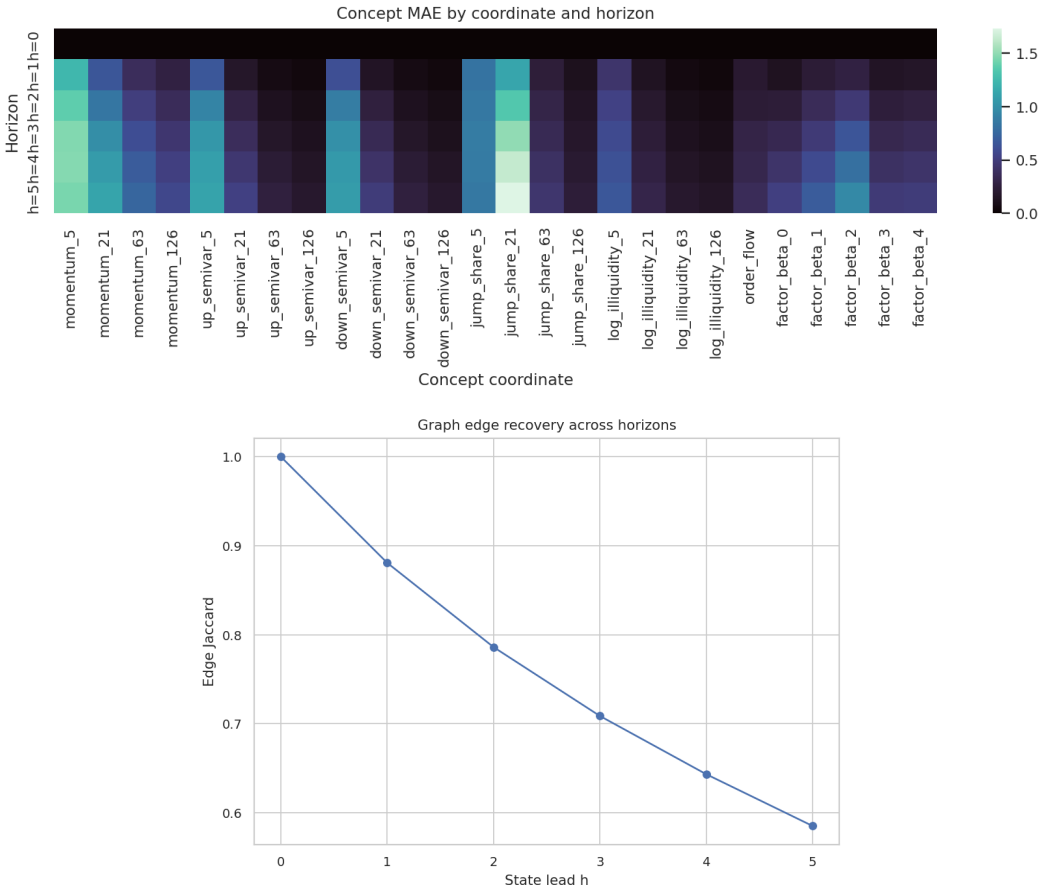


Figure 6: Concept and graph-path diagnostics for the calibrated fold-1, seed-1 run. Top: robust-standardized absolute concept error, averaged over origins and assets. Bottom: thresholded graph-support overlap by lead. Zero-lead agreement is anchored input agreement; subsequent columns/points describe predicted future states. These measurements test the state interface against its explicitly defined auxiliary targets.

It reports the Frobenius norm of the change in the complete five-lead mean tensor divided by the perturbation magnitude. The market state and initial graph are fixed. This is a realized directional response of the calibrated forecast; shrinking its mean also shrinks this diagnostic. It is not evidence that the recurrent state map has a smaller worst-case gain.

E.7 Efficiency and computational accounting

The training stage includes forward/backward optimization, complete post-epoch train/validation metrics, checkpoint selection, and optional scalar calibration. The archived prediction timer additionally encloses lightweight diagnostic-model scoring and the three input interventions; dividing it by the number of test origins would therefore mislabel diagnostic work as inference latency. These stages exclude the subsequent theory audit and do not measure peak device memory. Cached state preparation is shared across seeds, while cold preprocessing includes rolling factor regressions and Graphical-Lasso fits; the cached training timings must not be advertised as cold end-to-end runtimes.

For computational scaling, keep model widths fixed and consider one forecast origin with N assets, H leads, and S scenarios per lead. Dense graph/state operations require quadratic cross-sectional storage per lead, and dense precision inversion/Cholesky costs $O(HN^3)$ arithmetic. Applying the resulting factors to scenario shocks costs $O(SHN^2)$. Saving all scenarios requires $O(SHN)$ storage per origin. Here $O(\cdot)$ denotes asymptotic order up to fixed-width constants. Exact all-pairs ES

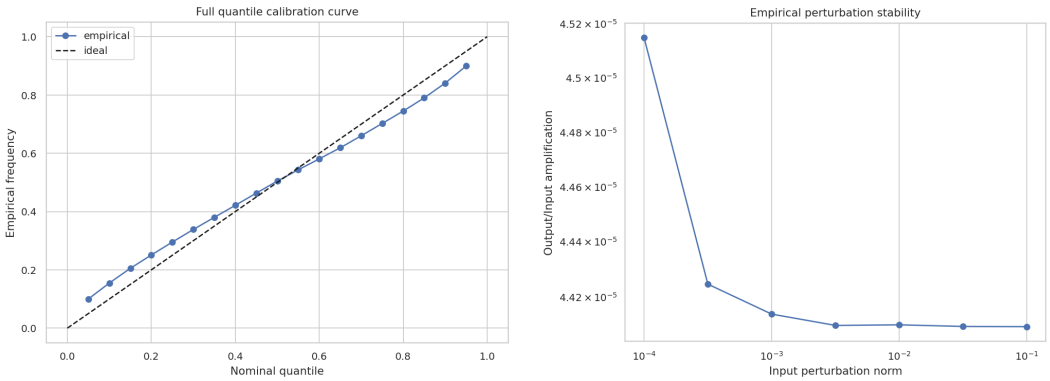


Figure 7: Different meanings of reliability, illustrated by the archived calibrated fold-1, seed-1 run. Left: empirical frequency below each forecast quantile versus its nominal probability; deviation from the diagonal shows imperfect distributional calibration. Right: mean-forecast perturbation amplification at one origin along one fixed random concept-space direction. Small directional response is not a worst-case Lipschitz bound and does not imply calibrated predictive uncertainty.

Table 12: Audit scope and observed outcomes across the 15 U.S. fitted state models. Calibrated/uncalibrated copies are not counted twice because mean shrinkage leaves the audited state dynamics unchanged. Numerical checks corroborate implementation; the analytic arguments establish the structural properties.

Property	Actual audit domain	Observed outcome
State feasibility	First test origin; predicted five-step path	Graph symmetry/diagonal errors 0; box violation 0
Positive precision	First test origin; all five leads	Minimum eigenvalue across runs ≥ 0.1048
Conditional risk response	First test origin; one risk coordinate; 17 fixed-graph decoder interventions	All slopes positive; minimum $\geq 3.01 \times 10^{-4}$
Local sensitivity estimate	8 origins per train/validation/test split, 5 leads, 15 models; 4 power iterations	1,800 states; largest estimated block-matrix radius 0.9740
Structural sensitivity summary	Conservative registered-box constants after fitting	Radius range 43.36–60.65; no implication of decaying errors
Calibration translation	Archived validation scenarios; first scenario pair checked numerically	Maximum pairwise-difference discrepancy 5.56×10^{-17}

evaluation has an additional $O(S^2HN)$ distance-computation cost per origin. The implementation uses dense operations even when measured edges are small, so sparse graph interpretation must not be confused with a sparse linear-algebra speed guarantee. Scalability to substantially larger asset universes requires a separate benchmark.

E.8 Cross-domain refitting on BTCF

A distinct financial measurement regime. The BTCF artifact distributed with FinTSBridge (Wang et al., 2025b) contains hourly observations of Bitcoin spot and perpetual futures. We use its two price/volume series and official chronological block boundaries, reformulated as future excess log-return prediction. The sole observed factor is the equal-weight contemporaneous spot/futures log return, denoted BTC_COMMON in the data. It is used only up to the current origin when measuring betas and residuals. The risk-free series is explicitly zero. Unlike the daily flow proxy, the flow coordinate uses twice taker-buy base volume divided by total base volume, minus one, for each recorded interval. This is still interval-aggregated flow, not a limit-order-book representation.

Table 13: Archived wall-clock stages in seconds, mean ± sample standard deviation over the 15 U.S. fold–seed settings on RTX 4090. Cross-fold variability includes different numbers of origins and selected epochs. Stage totals are not matched baseline speed comparisons.

Stage	Uncalibrated	Calibrated
Training, epoch evaluation, selection	74.15 ± 10.69	77.98 ± 11.24
Prediction, scoring, diagnostic input interventions	34.35 ± 4.75	34.31 ± 4.86

Table 14: BTCF refitting experiment, seed 1; all values are in raw units. ES is two-dimensional per lead, unlike the 32-dimensional U.S. task. No multi-seed uncertainty is implied.

Model	MSE	RMSE	MAE	ES	msIC	msIR
TREFIN	0.00003055	0.005527	0.003624	0.003833	0.002470	0.005015

The official preprocessed artifact does not recover original absolute price levels; the adapter exponentiates relative log prices so their successive log differences match the supplied series. Absolute dollar volumes therefore contain an unavailable normalization constant and are not directly comparable across the two instruments. The source has 73 non-unit-hour gaps, which are retained as consecutive observed-sequence steps rather than imputed. Thus a five-step prediction spans five subsequent records and need not always span exactly five clock hours. These boundaries prevent presenting the experiment as a new raw-tick or fully point-in-time microstructure benchmark.

Configuration and results. The model is refitted from scratch with seed 1. With one factor and four concept lookbacks, it has 22 asset-level concept coordinates and 56,086 trainable parameters. Concept lookbacks remain 5/21/63/126 observations; the graph lookback is 512, the market lookback is 21, and the graph is refitted at every observation. Hidden/lead/graph-embedding widths are 48/8/16. Batch size is 256, the epoch ceiling is 12, and patience is four; the remaining transition constants and loss weights follow the principal configuration. No post-training mean-shrinkage step is used in this archived run. The best validation checkpoint is epoch 10 of 12 completed epochs. Training-stage time is 266.09 seconds on the archived GPU setup.

The usable train/validation/test origin counts are 29,593/4,298/8,597. The test block begins 2023-12-08 03:00 and ends 2024-11-30 23:00, with origins purged at its end to keep all five labels in the block. Table 14 reports the single run without artificial seed deviations. Because this changes the universe, factor construction, sampling frequency, and width, it establishes that the measurement contract can be instantiated in another domain. It does not isolate a transfer benefit, demonstrate zero-shot generalization, or justify comparison to published price-level metrics on the original task.

Evidence scope. The main-table ranking, the rolling robustness results, the graph-free pilot, and the BTCF refit answer different questions and are not pooled into a single claim of state-of-the-art generalization. Full multi-seed structural ablations, matched baseline compute accounting, held-out concept-persistence comparisons, wider architectural sensitivity sweeps, and additional task types remain distinct empirical tests. The completed results establish competitive accuracy together with an explicit, testable state interface; they do not replace those tests with theoretical guarantees.