A MULTIVARIATE FINANCIAL TIME SERIES FOUNDATION MODEL WITH TEST-TIME LEARNING RELATIONAL ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate forecasting can improve financial forecast accuracy by drawing on related time series that provide predictive information beyond the target series' own history. However, financial time series are noisy, cross-series predictive relationships are non-stationary, and datasets contain different combinations of heterogeneous variables. We introduce FTFM, a financial time-series foundation model that couples causal temporal attention with Test-Time Learning Relational Attention. Causal temporal attention encodes each series' history. Relational attention then combines a slow path that weights these features by learned compatibility with a fast path guided by a directed predictive relation memory. Feature compatibility can remain high even as predictive relationships weaken. The relation memory adds direct feedback on how well temporal features across series predict subsequent observed features, using the resulting prediction errors to revise the relationships guiding cross-series attention during inference. Furthermore, similar normalized trajectories can represent different meanings and forecasting implications, we introduce semantic embeddings that preserve distinctions between series such as prices, returns, and volatility after normalization, while shared projections and sampling-interval conditioning support different series combinations and temporal resolutions. In empirical experiments across eight financial, economic, and synthetic domains, FTFM achieves the lowest forecasting errors among the evaluated models on the majority of benchmark tasks, both before and after LoRA adaptation.

1 INTRODUCTION

Financial time-series forecasting provides quantitative inputs for portfolio allocation, risk management, and economic decision-making. Multivariate forecasting can draw on information from related assets, markets, and economic variables beyond a target series' own history. However, financial observations are noisy, predictive relationships change over time, and datasets differ in their variables and sampling intervals.

Pretrained forecasting models offer a way to reuse representations across datasets (Das et al., 2024; Woo et al., 2024). Applying them to heterogeneous financial data requires both adapting cross-series information exchange and preserving the meaning of the input variables. Ordinary cross-series attention computes compatibility from history-conditioned features. Predicting subsequent features within the observed history provides an additional learning signal: prediction errors can revise the relations guiding cross-series information exchange. A shared model must also distinguish series meaning: similarly normalized trajectories can represent prices, returns, or volatility, with different forecasting implications.

We introduce FTFM, a financial time-series foundation model with *Test-Time Learning Relational Attention*. Causal temporal attention supplies features to both cross-series attention and a directed predictive relation memory F. The memory predicts subsequent observed features from preceding features across series in a learned projection space, allowing predictive relations to be estimated locally from each observed context. Prediction errors revise its coefficients, which modulate cross-series attention scores and value aggregation. Ordinary and relation-modulated attention share pro-

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.

jections and form a bounded weighted combination. Pretraining jointly learns the temporal representations and attention parameters through the forecasting objective. The same local learning rule operates during training and inference. During inference, pretrained parameters remain fixed while F is learned anew from the observed context, without future forecast labels.

FTFM pairs causal normalization with semantic embeddings of each series' supplied quantity family, field role, and representation. Normalization makes numerical scales comparable, while these embeddings distinguish what the values represent. Sampling-interval conditioning supplies the nominal duration of each time step. Shared patch projections and forecasting layers accommodate different numbers and combinations of series without pretrained instrument or series-pair parameters. The same pretrained parameters can therefore process different financial panels, with relations learned locally for each context.

We evaluate FTFM across eight financial, economic, and synthetic domains against pretrained and supervised forecasters. The pretrained models are evaluated first without downstream fitting and then after low-rank adaptation (LoRA) on the same benchmark training split. FTFM ranks first in a majority of domain–horizon comparisons for each error metric, both before and after LoRA adaptation, with 27.42 million stored pretrained parameters. The final pretraining run took 72 GPU-hours on a single NVIDIA RTX PRO 6000 GPU. Ablations assess the contributions of prediction-error correction, relational attention, selective expert routing, and semantic conditioning. Our contributions are:

- **Test-Time Learning Relational Attention:** an attention mechanism that learns directed predictive relations from observed transitions of causal temporal features and uses them to guide cross-series attention scores and value aggregation, with pretrained parameters fixed at inference.
- **A shared formulation for heterogeneous financial time series:** semantic conditioning and shared forecasting layers support different series identities and compositions while preserving supplied series meaning, with relations learned from each observed context.
- **Forecasting accuracy with a compact model:** lowest errors in a majority of the evaluated domain–horizon comparisons under each metric, both before and after LoRA adaptation, with 27.42 million stored pretrained parameters.

2 Related Work

Multivariate time-series forecasting. Multivariate forecasters differ in how they combine temporal information and cross-series dependencies. LTSF-Linear (Zeng et al., 2023) and PatchTST (Nie et al., 2023) provide competitive baselines without explicit cross-channel interaction. Crossformer separates temporal and cross-variable attention (Zhang & Yan, 2023), whereas iTransformer applies attention to tokens representing individual series (Liu et al., 2024). The Temporal Fusion Transformer uses variable selection and static covariate encoders to combine heterogeneous predictors (Lim et al., 2021). In Crossformer and iTransformer, cross-series attention already depends on the observed inputs. Other approaches use multiscale representations (Chen et al., 2024; Wang et al., 2024; 2025) and language-model adaptation (Cao et al., 2024; Jin et al., 2024).

Pretrained forecasting models. TimesFM and Chronos establish shared pretraining for forecasting unseen datasets without downstream fitting (Das et al., 2024; Ansari et al., 2024). The original Moirai accommodates variable numbers of series (Woo et al., 2024), while Timer-XL formulates multivariate forecasting as next-token prediction (Liu et al., 2025c). Toto separates temporal and variate attention and uses patch-based normalization (Cohen et al., 2025). Chronos-2 combines temporal attention with attention within series groups, including known-future covariates (Ansari et al., 2025). Such multivariate architectures support cross-series conditioning with frozen pretrained parameters. Architectural choices also include compact mixer networks in Tiny Time Mixers (Ekambaram et al., 2024) and sparse expert activation in Time-MoE and Moirai-MoE (Shi et al., 2025; Liu et al., 2025b).

Financial forecasting and series semantics. HIST combines predefined and learned stock concepts (Xu et al., 2021), and MASTER models changing cross-stock interactions for return prediction (Li et al., 2024). Financial applications include generative modeling (Huang et al., 2024; Li

et al., 2025; Soun et al., 2026) and forecasting with verbal reasoning (Koa et al., 2026). Financial foundation models make different choices about joint conditioning: FinCast forecasts channels independently with shared weights (Zhu et al., 2025), whereas Kronos jointly tokenizes an instrument's price, volume, and turnover fields (Shi et al., 2026). LENS models temporal and cross-channel dependencies (Xu et al., 2025); DELPHYNE supports joint forecasting across instruments (Ding et al., 2025). Pretraining across assets and conditioning jointly on them are thus distinct design choices. FTFM uses supplied quantity-family, field-role, and representation metadata to distinguish series meaning from numerical scale after normalization, with forecasting layers shared across panel compositions without pretrained instrument identifiers.

Adaptive cross-series dependencies. LIFT estimates local leading indicators and aligns their lags to refine forecasts (Zhao & Shen, 2024). DeltaLag uses sparse cross-attention to select leading stocks and pair-specific lags for return prediction (Zhou et al., 2025). CoRA uses dynamic correlation estimates to guide contrastive fine-tuning of foundation-model adapters (Cheng et al., 2026). Dynamic Relational Priming conditions attention keys and values on interaction partners (Lee & Clark, 2026). FTFM learns relations through a different local task: predicting subsequent observed temporal features across series and using the resulting errors to revise the predictor within each context.

Test-time learning and predictive relation memory. Delta-rule fast-weight learning updates associative memory through prediction errors (Schlag et al., 2021); DeltaNet parallelizes these updates over sequence length (Yang et al., 2024), and Gated DeltaNet combines them with gated forgetting (Yang et al., 2025). TTT layers treat recurrent states as models learned through per-sequence self-supervision (Sun et al., 2025), while Titans combines attention with a neural memory trained to retain historical information (Behrouz et al., 2025). Test-Time Learning Relational Attention builds on these learning principles with a directed predictive relation memory indexed by source and target series. Causal temporal features support both its local prediction task and cross-series attention; learned relation strengths modulate attention scores and value aggregation through shared projections and a bounded weighted combination. Pretraining jointly learns the feature space and update parameters for forecasting, while inference reconstructs the memory within each context with pretrained parameters fixed.

3 METHOD: FTFM

3.1 PROBLEM FORMULATION

A *panel* comprises scalar financial time series aligned on a common logical time axis. Its T-step context is $X \in \mathbb{R}^{T\times S}$, where S counts scalar series and may include multiple fields per instrument. Inputs also include an observation mask $M \in \{0,1\}^{T\times S}$, a time-validity mask $\tau \in \{0,1\}^{T}$ identifying real time steps, nominal cadence $\Delta_{\mathrm{nom}} > 0$, and per-series categorical metadata. Equations omit padded series, which are masked throughout. FTFM predicts marginal quantiles over a fixed direct horizon $H_{\max}$ and a separate point forecast for the next patch. Pretrained forecasting and update parameters θ are shared across series identities, compositions, and widths.

3.2 CAUSAL PANEL REPRESENTATION

FTFM uses causal normalization, semantic embeddings, and cadence conditioning to address differences in numerical scale, series meaning, and sampling interval, respectively.

The context is divided into $L = T/P$ non-overlapping patches of P steps. For series s at patch ℓ, let $\mu_{\ell s}$ and $\sigma_{\ell s} > 0$ be the masked prefix mean and scale, using observations through the patch endpoint. Their definitions and boundary cases are given in Appendix C.1. For observed $t \in \mathcal{P}_\ell = \{(\ell-1)P+1,\dots,\ell P\}$, the transformed values and patch features are

$$u_{ts} = \operatorname{asinh}\left(\frac{x_{ts}-\mu_{\ell s}}{\sigma_{\ell s}}\right), \qquad \phi_{\ell s} = \left[u_{\mathcal{P}_\ell,s}; M_{\mathcal{P}_\ell,s}; \tau_{\mathcal{P}_\ell}\right] \in \mathbb{R}^{3P}. \tag{1}$$

The asinh transform compresses large standardized magnitudes while preserving their sign. Missing and padded cells have transformed value zero, distinguished from observed zeros by the masks. Prefix statistics preserve patch-boundary causality without dataset-fitted statistics.

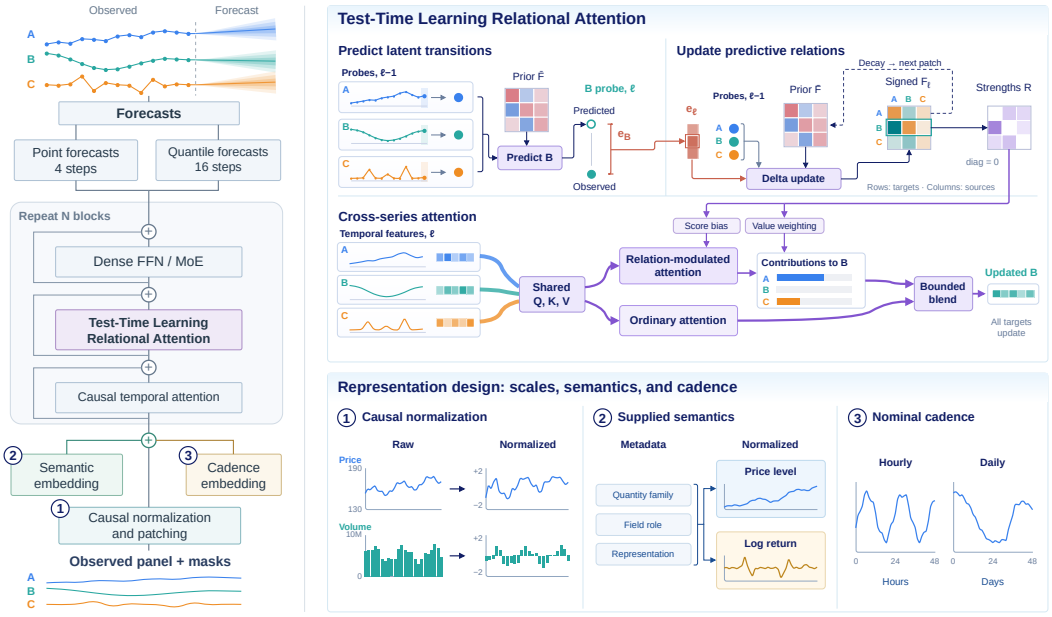


Figure 1: FTFM architecture. **Left:** Input conditioning, repeated backbone blocks, and point and quantile forecast heads. **Upper right:** Test-Time Learning Relational Attention uses prediction errors to update the directed predictive relation memory F. Derived strengths R modulate attention scores and value aggregation; ordinary and relation-modulated attention share projections and form a bounded weighted combination. **Lower right:** Causal normalization, semantic embeddings, and cadence embeddings address numerical scale, series meaning, and sampling interval, respectively.

The input embedding combines a shared patch projection with cadence and semantic terms:

$$h_{\ell s}^{(0)} = E_{\text{patch}}(\phi_{\ell s}) + E_{\text{sem}}(q_s, f_s, r_s) + E_{\text{cad}}(\Delta_{\text{nom}}). \tag{2}$$

The descriptors q_s, f_s, r_s identify quantity family, field role, and representation; their parameterization is given in Appendix C.1. They distinguish similarly normalized price levels, returns, and volatility without instrument-identity or series-position embeddings. Volatility descriptors supply meaning rather than estimate current volatility.

3.3 Backbone Overview

The decoder-only backbone (Figure 1) preserves $h \in \mathbb{R}^{L \times S \times D}$. Each block applies pre-normalized residual sublayers for causal temporal attention, same-patch Test-Time Learning Relational Attention, and a feed-forward network, in that order. Temporal attention accesses only patches $m \leq \ell$ within each series and supplies features to both the probes and cross-series attention projections. Dense and mixture-of-experts feed-forward sublayers alternate, with each mixture-of-experts sublayer selecting two of six experts per token. Appendix C.2 specifies the block configuration.

Wholly padded patches and series are excluded from attention; numeric missingness does not invalidate real tokens. Shared projections and feed-forward parameters make inference equivariant to joint permutations of series, masks, and metadata, with stored parameter count independent of panel width.

3.4 Test-Time Learning Relational Attention

Test-Time Learning Relational Attention fits a directed predictive relation memory F to observed transitions between temporal features. Prediction errors update F, whose learned relations modulate cross-series attention.

In each block, a shared linear projection of normalized post-temporal hidden states produces one scalar probe per head and series. Suppressing block and head indices, let $z_\ell \in \mathbb{R}^S$ collect these

probes and $F_\ell \in \mathbb{R}^{S\times S}$ denote the state after patch ℓ. Entry $F_{\ell,sr}$ represents a lag-one predictive coefficient from source r to target s. The probes summarize temporal features, so this one-patch transition can incorporate information from earlier observations. Each panel, block, and head starts with $F_1 = 0$.

For $\ell \geq 2$, the local objective is $\mathcal{J}_\ell(F) = \frac{1}{2}\|z_\ell - Fz_{\ell-1}\|_2^2$. The update applies decay followed by a prediction-error correction normalized by source energy:

$$\begin{aligned} \bar{F}_\ell &= \exp(-\rho\Delta_\ell)F_{\ell-1}, \qquad \Delta_\ell = P\Delta_{\text{nom}}, \\ e_\ell &= z_\ell - \bar{F}_\ell z_{\ell-1}, \\ F_\ell &= \bar{F}_\ell + g_\ell\eta\frac{e_\ell z_{\ell-1}^\top}{\epsilon + \|z_{\ell-1}\|_2^2}. \end{aligned} \tag{3}$$

Each block and head learns a step size $0 < \eta < 1$ and decay rate $\rho > 0$; $\epsilon > 0$ stabilizes the denominator. The gate $g_\ell = 1$ only when all P positions of every real series are time-valid and observed at both transition endpoints; otherwise the state only decays. The predictor includes self-lags through the diagonal of F.

Equation 3 is a normalized gradient step on $\mathcal{J}_\ell$ after decay, following established delta-rule mathematics (Schlag et al., 2021; Yang et al., 2024); the derivation is in Appendix C.3. During pretraining, forecasting gradients pass through causal temporal attention, the probe projection, the recurrence, and cross-series attention, jointly learning the feature space and update, decay, and mixing parameters. At inference, the same rule updates only F, with θ fixed. The local loss defines the correction, not an additional forecasting-loss term.

The transition ending at patch ℓ updates F_ℓ before same-patch attention uses it. All local targets lie within the observed prefix, including its latest transition, preserving causality at patch boundaries.

For one of A heads, both attention paths share $Q_\ell, K_\ell, V_\ell \in \mathbb{R}^{S\times d}$, with $d = D/A$. Their projections and the separate probe projection use the same normalized post-temporal features. We convert the signed memory into bounded off-diagonal strengths and form ordinary and relation-modulated attention maps:

$$\begin{aligned} R_{\ell,sr} &= \mathbf{1}[s \neq r]\,|\tanh(F_{\ell,sr})|, \qquad C_\ell = \frac{Q_\ell K_\ell^\top}{\sqrt{d}}, \\ A_\ell^{\text{slow}} &= \text{softmax}(C_\ell), \qquad\qquad A_\ell^{\text{fast}} = \text{softmax}(C_\ell + \beta R_\ell). \end{aligned} \tag{4}$$

Coefficients of either sign can strengthen attention; the zero diagonal restricts relation modulation to cross-series exchange. Both row-wise softmaxes use the same source mask and include the valid self key. The paths form a bounded weighted combination:

$$O_\ell = \left[(1-\lambda_{\text{eff}})A_\ell^{\text{slow}} + \lambda_{\text{eff}}\left(A_\ell^{\text{fast}} \odot R_\ell\right)\right] V_\ell. \tag{5}$$

The logit scale β and mixing coefficient λ are learned, bounded positive scalars for each block and head (Appendix C.3). For a real query, $\lambda_{\text{eff}} = \lambda$ if another real series exists, and zero otherwise. Without renormalization of $A_\ell^{\text{fast}} \odot R_\ell$, relation strengths control both source weighting and total aggregation weight. "Slow" denotes ordinary attention and "fast" its relation modulation; both depend on the current temporal features. The coefficients of F describe directed predictive associations in probe space, not causal effects or calibrated measures of forecasting usefulness. Appendix B.1 illustrates these updates and their effect on attention in a real benchmark context.

3.5 Forecast Heads and Training Objective

Training supervises eligible prefixes from $h_2, \ldots, h_L$ in one backbone pass. A shared point head provides next-patch supervision and targets accurate financial forecasts over one to four steps. A shared quantile head predicts marginal quantiles on a fixed grid $\mathcal{Q}$ over $H_{\max}$ steps, providing probabilistic forecasts. Targets and losses use the asinh-transformed space with the statistics $(\mu_{\ell s}, \sigma_{\ell s})$ of their origin ℓ.

Let $\mathcal{E}$ contain eligible prefixes with available targets for both heads. For $i \in \mathcal{E}$, $\mathcal{L}_{\text{p},i}$ and $\mathcal{L}_{Q,i}$ average Huber and pinball losses over available targets, respectively, with the latter also averaging over $\mathcal{Q}$.

The objective is

$$\mathcal{L} = \frac{1}{|\mathcal{E}|} \sum_{i \in \mathcal{E}} (\alpha_{\mathrm{p}} \mathcal{L}_{\mathrm{p},i} + \alpha_{\mathrm{q}} \mathcal{L}_{Q,i}) + \alpha_{\mathrm{b}} \mathcal{L}_{\mathrm{balance}} + \alpha_{\mathrm{z}} \mathcal{L}_{z}. \tag{6}$$

Each eligible prefix has equal outer weight regardless of panel width. The router terms regularize expert use and routing logits. The fixed coefficients, eligibility rules, and complete head and loss definitions are given in Appendix C.4.

Inference decodes the final context state and restores raw units with $\hat{x} = \mu_{Ls} + \sigma_{Ls} \sinh(\hat{u})$. The $q = 0.50$ quantile supplies the median for $H \leq H_{\max}$; the Huber-trained point prediction is separate. Marginal quantiles do not specify a joint distribution across future times and series. Longer-horizon benchmarks use recursive median conditioning under the protocol in Appendix D.

4 Experiments

We evaluate forecasting accuracy across heterogeneous financial series both without downstream parameter fitting and after supervised LoRA adaptation. The first setting assesses performance using the pretrained models directly; the second tests whether FTFM remains competitive when all pretrained competitors can also learn from the benchmark training split. Both settings use the same domain-dependent context policy.

4.1 Experimental Setup

FTFM pretraining. We pretrain the 27.42M-parameter FTFM on a diverse collection of financial and other time series, comprising approximately 6.4 billion scalar observations. We explicitly separate the benchmark dataset from the pretraining dataset during construction and ensure that the pretraining dataset contains no future information from the benchmark dataset. The forecasting objective in Section 3.5 jointly optimizes the shared representations and parameters governing context-local relation learning, while the directed predictive relation memory is learned within each context. Optimization uses AdamW with a peak learning rate of 2×10^{-4}, weight decay 0.1, and gradient-norm clipping at 1. Computation uses BF16, while model parameters and optimizer states remain in FP32. The final pretraining run required 72 GPU-hours on a single NVIDIA RTX PRO 6000 GPU. Appendix D provides the model configuration and complete optimization schedule.

Data and forecasting protocol. The benchmark spans eight domains: Crypto, Equities, Futures, Exchange, M4 Finance, Economy, FTS mixed, and Synthetic, with panels of 1–50 scalar series. It contains 61.1 million scalar entries: 40.2 million for training, 10.4 million for validation, and 10.5 million for testing. Non-M4 data are split chronologically 60/20/20 within each source trajectory's selected span. M4 retains its official test horizon and reserves the final horizon-length segment of its official training history for validation. Forecasts use only preceding observations, with short histories padded and masked. Non-M4 forecast horizons are 4, 16, and 32 sampling steps; for M4, we follow the original forecast horizons used in the M4 Competition (Makridakis et al., 2020). Appendix D details the sources, splits, and horizon mapping.

Models and inference. We select pretrained competitors for their strong reported forecasting performance, including state-of-the-art results on established benchmarks. Chronos-2 (Ansari et al., 2025), TimesFM 3 (Jain & Sen, 2026), Moirai2 (Liu et al., 2025a), and Toto2 (Khwaja et al., 2026) provide general-purpose pretrained comparisons, while FinCast (Zhu et al., 2025) provides a finance-specific counterpart. TSMixer (Chen et al., 2023), PatchTST (Nie et al., 2023), and iTransformer (Liu et al., 2024) complement these models with established supervised architectures spanning MLP mixing, channel-independent temporal attention, and cross-series attention, respectively. AR(1) provides a simple statistical reference. Appendix D.5 discusses additional financial foundation models and explains their omission on grounds of public checkpoint availability or input compatibility.

All six pretrained models use context 128 for Crypto, Economy, and FTS mixed; 64 for Equities, Exchange, and M4 Finance; and 32 for Futures and Synthetic. This policy is shared by the base and adapted comparisons. In the base comparison, pretrained parameters remain fixed during forecasting. AR(1) uses a validation-selected configuration refit on training and validation data. Each

neural baseline uses one global model per seed, trained for 3,000 updates on the training split with context 128. Model-specific settings are selected to maximize overall validation performance under a consistent tuning and evaluation protocol, providing each method with a fair and competitive configuration. Architecture, optimization, channel handling, and checkpoint details appear in Appendix D.

Metrics and reporting. We report mean absolute scaled error (MASE) and symmetric mean absolute percentage error (sMAPE), both used in the M4 Competition (Makridakis et al., 2020). MASE measures error relative to historical one-step changes, while sMAPE measures error relative to observed and predicted magnitudes. A piece is a contiguous row span of a source panel. For each piece–series–horizon with n valid origin–lead pairs, we compute $\mathrm{MASE} = \frac{1}{na^\star}\sum_i e_i$ and $\mathrm{sMAPE} = \frac{200}{n}\sum_i \frac{e_i}{|y_i|+|\hat{y}_i|}$, where $e_i = |y_i - \hat{y}_i|$ is the absolute error in original units. The scale $a^\star$ is derived from the mean absolute lag-one change in pre-test history, independently of the input context, for all domains including M4. Piece–series scores receive equal weight within each domain–horizon row. Lower values are better; sMAPE is reported as a percentage. Appendix D specifies degenerate cases and run averaging.

4.2 Main Forecasting Results

Table 1 shows broad competitiveness without downstream fitting. FTFM achieves the lowest MASE in 19 of 26 settings and the lowest sMAPE in 15; its remaining MASE results are within 5.9% of the best competing model. The sMAPE reductions are smaller on Equities and Futures, where the advantage is more pronounced under historical-variation scaling. FTFM retains the lowest errors under both metrics in five domains at 32 steps, showing that its relative competitiveness persists when forecasting requires recursive conditioning beyond its sixteen-step forecast head. These domains include the univariate Exchange and Economy settings, where information exchange between distinct series is unavailable. The results thus support the complete forecasting model.

Table 1: Base forecasting comparison. **Best**; second best.

Domain	h	FTFM		Chronos-2		TimesFM 3		FinCast		Moirai2		Toto2		AR(1)		TSMixer		PatchTST		iTransformer	
		MASE	sMAPE	MASE	sMAPE	MASE	sMAPE	MASE	sMAPE	MASE	sMAPE	MASE	sMAPE	MASE	sMAPE	MASE	sMAPE	MASE	sMAPE	MASE	sMAPE
Crypto	4	**1.0490**	**13.5130**	1.3104	15.4202	1.2596	15.0606	1.3786	17.6538	1.8486	18.7678	1.2432	15.0598	1.5674	19.7133	1.2642	16.5856	1.2938	17.2453	1.2714	16.8532
	16	**2.1509**	**14.6425**	2.5044	16.1459	2.5699	16.1844	2.4850	18.3691	3.4788	19.9916	2.4228	15.9590	3.5926	20.4380	2.4805	19.2025	2.5134	20.4336	2.4668	18.7484
	32	**2.7189**	**15.1464**	3.0741	16.8099	3.2558	17.0245	2.9785	18.7861	4.3875	22.4125	3.0466	16.8388	5.3625	22.0192	2.9604	21.0117	3.0842	21.9713	2.9428	19.9513
Equities	4	**1.0865**	**18.1018**	1.1822	18.5506	1.1964	18.3872	1.2454	20.7104	1.2806	20.2106	10.7240	18.4849	1.2859	19.5826	1.1540	19.9232	1.1697	20.1593	1.1801	20.6414
	16	**1.9317**	**18.2146**	2.1019	18.6542	2.1934	18.4968	2.0621	20.9098	2.1658	21.0649	11.5736	18.6130	2.5731	19.5968	1.9967	20.3541	2.0320	21.5194	2.0086	20.9621
	32	**2.6917**	**18.4487**	2.8939	19.0011	3.1833	18.6915	2.7940	21.3326	2.9742	22.9429	12.2228	18.8630	3.9246	19.6390	2.7825	21.3278	2.8456	22.3199	2.7854	22.3260
Futures	4	**1.1910**	**15.8897**	1.3229	16.1366	1.3721	16.4817	1.3839	19.4054	2.5958	17.2286	1.2906	16.0456	1.6347	20.0673	1.2719	18.2880	1.2849	19.2184	1.2853	19.1777
	16	**1.8408**	**16.0677**	2.0638	16.3655	2.1988	16.7204	2.0125	19.7352	3.5267	17.6263	2.0644	16.1600	3.0581	20.3759	1.9485	19.1906	1.9923	20.9473	1.9502	19.9256
	32	**2.4583**	**16.3714**	2.7117	16.6364	3.0529	17.0322	2.6238	20.1315	4.8832	18.4735	2.7581	16.5526	4.4808	20.4682	2.6034	20.1987	2.6743	22.1913	2.5808	21.4208
Exchange	4	1.9513	0.6097	1.2135	0.6617	2.0545	0.6403	2.0433	0.6391	2.0257	0.6324	2.0278	0.6325	1.9673	0.6159	1.9679	0.6152	**1.9447**	**0.6078**	1.9463	0.6083
	16	**3.0873**	**0.9603**	3.4422	1.0609	3.4649	1.0799	3.2306	1.0070	3.3775	1.0399	3.3365	1.0371	3.2053	1.0041	3.2036	1.0049	3.1867	0.9990	3.1314	0.9812
	32	**4.0169**	**1.2374**	4.4634	1.3638	4.4197	1.3848	4.1071	1.2712	4.3248	1.3240	4.2821	1.3142	4.3115	1.3464	4.3008	1.3320	4.1928	1.3065	4.1199	1.2823
M4 Finance	6	3.4934	15.3849	3.5284	16.4397	**3.3874**	15.4509	4.8821	20.8231	3.9687	17.4281	3.5987	16.3517	3.7389	15.9359	3.4924	15.2155	3.4650	**15.2111**	3.5169	15.3117
	8	2.3654	12.6846	2.3763	11.9379	2.3488	11.8817	3.2176	15.7652	2.3335	11.9348	**2.2460**	**11.4285**	2.7029	14.9232	2.6782	14.3129	2.5615	12.8946	2.7486	14.0623
	14	**3.3634**	**3.3719**	3.6443	3.5442	3.6704	3.5136	3.7132	3.7440	3.6441	3.5715	3.4236	3.3733	3.6554	3.6423	3.4400	3.5147	3.5146	3.5395	3.4359	3.4765
	18	3.3445	14.6192	3.1924	13.9479	3.3130	14.4031	3.9978	16.5672	**3.1587**	13.9088	3.2115	**13.8936**	3.7772	16.9121	3.6677	18.9848	3.5365	15.4045	3.6829	17.6902
Economy	4	2.2891	8.4405	2.3843	8.6272	2.2850	8.5144	3.0152	10.6169	2.4049	8.6923	**2.2620**	**8.3536**	3.1013	9.9178	2.5302	9.1719	2.4882	8.8277	2.5755	9.2128
	14	**2.9639**	**2.4042**	3.2872	2.5260	3.5019	2.8192	3.4820	2.6973	3.1874	2.5530	3.2841	2.5919	3.6330	3.0564	3.3152	2.7048	3.2586	2.6457	3.2701	2.6239
	16	**4.4021**	**12.0846**	4.6267	12.8664	4.8206	13.2669	5.8523	14.9011	4.5011	12.7413	4.6289	12.5565	8.1061	15.4966	5.6750	14.7994	5.4841	13.7465	5.8519	14.7780
	32	**4.6850**	**16.0633**	5.4734	18.1111	5.3645	17.4788	7.0657	18.2726	4.8452	16.7103	5.8270	17.2773	10.2801	19.1428	6.0420	16.8033	6.0231	16.7178	6.3109	16.9956
FTS mixed	4	**1.5007**	22.2958	2.0416	22.8786	4.5123	**16.2922**	2.1484	24.0390	3.6321	25.3620	33.6243	27.4442	2.1115	19.2420	1.7289	18.7896	1.6879	18.6240	1.7025	18.2976
	16	3.0871	24.2270	6.5761	25.0632	7.1390	**18.4595**	6.7286	26.1858	8.7849	28.5182	47.8590	31.1620	3.8610	22.3313	3.0721	21.6543	3.0579	21.9595	**3.0422**	20.8789
	32	4.4842	25.6278	7.9514	26.4650	9.0249	**20.3784**	8.1284	27.8261	10.5070	30.5400	51.0004	32.5448	36.0268	24.3687	4.2665	24.0539	4.3109	24.6380	**4.2495**	23.5228
Synthetic	4	**0.8808**	41.9215	0.8980	41.7719	0.9022	41.2668	1.0169	**39.3682**	0.9275	41.5751	0.8927	41.8670	0.8918	45.8511	0.9308	39.9581	0.9061	40.6136	0.9220	40.3641
	16	**0.9738**	42.9183	0.9994	42.7825	1.0079	42.3337	1.1051	**40.4618**	1.0335	42.3925	0.9886	42.7094	1.0266	47.7107	1.0424	40.9628	1.0208	41.1736	1.0136	41.2376
	32	**1.0449**	43.1447	1.0734	43.1701	1.0962	42.7751	1.1771	**40.8755**	1.0958	42.7653	1.0610	43.0566	1.1456	48.3001	1.1554	41.3428	1.1215	41.2290	1.0945	41.4794
1st-place count		**19**	**15**	**0**	**0**	**1**	**3**	**0**	**3**	**1**	**0**	**2**	**3**	**0**	**0**	**0**	**0**	**1**	**2**	**2**	**0**

4.3 LoRA Adaptation

Each pretrained model is adapted across the benchmark training domains using LoRA, with all original weights and biases frozen. Attention adapters use rank 2 and scaling parameter $\alpha = 4$; forecast-head adapters use rank 8 and $\alpha = 16$. All use training context 128 and learning rates from 5×10^{-6} to 5×10^{-7}. We evaluate final adapter checkpoints. The objectives and model-specific

batching are detailed in Appendix D. We use the same evaluation settings as in the base forecasting comparison.

Table 2: Forecasting after LoRA adaptation. **Best**; second best.

Domain	h	FTFM		Chronos-2		TimesFM 3		FinCast		Moirai2		Toto2	
		MASE	sMAPE	MASE	sMAPE	MASE	sMAPE	MASE	sMAPE	MASE	sMAPE	MASE	sMAPE
Crypto	4	**1.0463**	**13.5249**	1.2386	14.9277	1.2498	15.0316	1.3505	16.9750	1.7155	18.0968	1.2205	14.8754
	16	**2.1569**	**14.6562**	2.4244	15.9749	2.5008	16.2007	2.4730	17.3507	3.0906	19.5946	2.3820	15.9514
	32	**2.7258**	**15.2317**	2.9332	16.4532	3.1204	16.8640	2.9493	17.8727	3.7551	21.7136	2.9102	16.5041
Equities	4	**1.0850**	**18.0862**	1.1262	18.4351	1.1572	18.4446	1.2061	19.6581	1.2758	20.0724	8.9345	18.4047
	16	**1.9288**	**18.1943**	1.9590	18.5235	2.0725	18.6681	2.0085	19.6376	2.1546	20.8631	9.1382	18.5653
	32	**2.6895**	**18.4150**	2.7266	18.8521	3.0555	18.8603	2.7693	19.7846	2.9615	22.7216	9.6154	18.8265
Futures	4	**1.1909**	**15.8943**	1.2756	16.3153	1.3553	16.5929	1.3381	18.0655	2.4335	17.1987	1.2607	16.2299
	16	**1.8411**	**16.1059**	1.9287	16.5019	2.1388	16.8321	1.9974	18.5627	3.1759	17.5800	1.9457	16.3132
	32	**2.4599**	**16.4128**	2.5547	16.7254	2.9944	17.0672	2.6055	18.8680	4.2015	18.4124	2.6069	16.6816
Exchange	4	**1.9428**	**0.6064**	1.9833	0.6191	2.0310	0.6337	2.0493	0.6390	2.0240	0.6316	1.9758	0.6165
	16	**3.0771**	**0.9563**	3.1724	0.9879	3.3228	1.0365	3.1964	0.9981	3.3709	1.0376	3.2164	0.9984
	32	3.9586	**1.2181**	4.0992	1.2676	4.2684	1.3320	**3.9526**	1.2296	4.3142	1.3199	4.0944	1.2561
M4 Finance	6	3.3814	14.9601	**3.1611**	**14.2709**	3.2268	14.6792	3.3428	14.7951	3.8414	16.9657	3.4590	15.3160
	8	2.2631	12.1579	**2.1541**	**11.0344**	2.2145	11.4481	2.5342	13.5805	2.3067	11.8481	2.2625	11.3889
	14	**3.2188**	3.2344	3.3227	3.3033	3.4965	3.5108	3.6235	3.6313	3.6119	3.5346	3.3564	**3.2261**
	18	3.1600	14.1329	**3.0562**	**13.6538**	3.1645	14.0553	3.4114	15.0454	3.1439	13.8708	3.1839	13.7873
Economy	4	2.2482	8.4860	2.2596	8.4417	2.2745	8.5355	2.6156	9.9044	2.3612	8.6367	**2.2294**	**8.3799**
	14	**2.9544**	**2.3801**	3.1326	2.4490	3.3618	2.6435	3.4055	2.7482	3.1862	2.5438	3.1761	2.4863
	16	**4.4431**	**12.2792**	4.5182	12.6044	4.6780	13.1304	5.1200	14.4003	4.4737	12.6810	4.6015	12.6421
	32	4.6162	**16.0010**	**4.5541**	16.0293	4.9403	16.6245	6.0655	17.1072	4.8365	16.6367	5.2192	16.5457
FTS mixed	4	**1.4959**	22.2609	1.9536	22.7029	4.2134	**16.2809**	2.1419	23.4777	3.3774	24.8668	20.8369	25.8701
	16	**3.2285**	24.1824	6.4457	24.8063	7.0027	**18.4910**	6.7103	25.5787	8.4391	27.9702	40.4252	30.6245
	32	**4.6252**	25.5604	7.7956	26.2430	8.8314	**20.3578**	8.0866	26.9058	10.0666	29.8992	43.0086	31.6582
Synthetic	4	**0.8806**	41.9426	0.8923	41.4779	0.9040	41.3786	0.9094	**40.3998**	0.9240	41.6246	0.8893	42.0099
	16	**0.9736**	42.9440	0.9869	42.5274	1.0116	42.2482	1.0080	**41.3162**	1.0287	42.4411	0.9836	42.8504
	32	**1.0445**	43.1874	1.0575	42.8581	1.0971	42.7222	1.0814	**41.7386**	1.0918	42.8148	1.0533	43.1718
1st-place count		**20**	**15**	**4**	**3**	**0**	**3**	**1**	**3**	**0**	**0**	**1**	**2**

Table 2 shows that across the four M4 Finance subsets, adaptation reduces MASE by 3.2–5.5% and sMAPE by 2.8–4.2%. LoRA also improves FTFM's overall forecasting performance. The adapted comparison also gives competing models an opportunity to improve their fit to the benchmark. Although their improvements narrow several performance gaps, FTFM retains the lowest errors under both metrics across Crypto, Equities, and Futures at every horizon. Its advantage persists after competing models receive the same benchmark training split, strengthening the evidence beyond the frozen-model comparison.

4.4 Summary of Results

FTFM combines strong held-out forecasting accuracy with 27.42 million stored pretrained parameters, achieving the lowest errors in a majority of the evaluated domain–horizon settings under each metric, both before and after LoRA adaptation (Tables 1 and 2). The gains vary across error measures: MASE scales errors by historical one-step variation, whereas sMAPE scales them by observed and predicted magnitudes. For example, after LoRA adaptation, FTFM leads MASE on FTS mixed at horizon 16 and on Synthetic at every evaluated horizon. It also achieves the lowest sMAPE across Crypto, Equities, Futures, and Exchange at every evaluated horizon. Its broad held-out competitiveness suggests that FTFM learns generalizable patterns in noisy financial series, where memorizing noise can hurt forecasts on new data. This performance is achieved with fewer stored parameters than four of the five pretrained competitors (Table 3). The compact parameterization reduces the memory required for model weights, supporting deployment on edge devices and GPUs with limited VRAM.

Table 3: Pretrained model size (millions of stored parameters).

FTFM	Chronos-2	TimesFM 3	FinCast	Moirai2	Toto2
27.42	119.48	330.71	991.44	11.39	312.68

5 Ablation Studies

For each training ablation, we train a separate model with the specified component removed or modified and compare its forecasting performance with the full model. Table 4 compares four separately trained ablations and inference-only semantic-embedding removal against full FTFM. All evaluations use the domain-specific context lengths and reported horizons of Table 1, whose full-model scores are reused. Domain scores average the reported horizons, and improvements average per-domain relative error reductions. Two-sided paired t-tests use the 26 domain–horizon error differences, treated as independent observations.

Table 4: Domain-average ablation results. p-values use two-sided paired t-tests of error differences.

Domain	Full model		Training ablations								Inference-only ablation	
	FTFM		No prediction-error correction		Ordinary series attention		All six experts		Trained without semantic embeddings		Semantic embeddings disabled at inference	
	MASE	sMAPE	MASE	sMAPE	MASE	sMAPE	MASE	sMAPE	MASE	sMAPE	MASE	sMAPE
Crypto	1.9729	14.4340	2.0442	15.1884	2.0629	15.0641	2.0276	14.8027	2.0029	14.9167	1.9859	14.9538
Equities	1.9033	18.2550	1.9061	18.5474	1.9086	18.4103	1.9073	18.3195	1.9053	18.4363	1.9026	18.4021
Futures	1.8300	16.1096	1.8434	16.5327	1.8486	16.4764	1.8394	16.3020	1.8382	16.4934	1.8331	16.4370
Exchange	3.0185	0.9358	3.0801	0.9556	3.1198	0.9661	3.0434	0.9428	3.0790	0.9557	3.0576	0.9463
M4 Finance	3.1417	11.5152	3.1313	11.4963	3.0921	11.4367	3.1204	11.3167	3.0821	11.3971	3.1425	11.5305
Economy	3.5850	9.7482	3.8168	10.0665	4.1276	10.4908	3.8515	10.1216	3.8183	9.9733	3.5834	9.7754
FTS mixed	3.0240	24.0502	4.3385	24.4834	4.0635	24.1122	3.2839	24.1343	3.9885	24.4031	4.9758	24.2264
Synthetic	0.9665	42.6615	0.9669	42.7735	0.9670	42.7503	0.9667	42.7716	0.9781	42.7050	0.9857	42.1495
Overall average	2.4303	17.2137	2.6409	17.5055	2.6488	17.4634	2.5050	17.3389	2.5865	17.4101	2.6833	17.3026
Mean FTFM improvement (%)	–	–	5.30	2.03	5.76	2.16	2.30	0.91	4.19	1.42	5.40	0.91
p-value	–	–	0.0498	0.0001	0.0149	0.0009	0.0196	0.0132	0.0557	0.0018	0.1392	0.1751

The mean paired error reductions over ordinary attention and the no-correction variant have 95% confidence intervals excluding zero on both metrics. Because both attention variants retain causal temporal features, these comparisons support learning predictive relations within each context and refining them through prediction-error feedback.

Selective routing also improves both metrics with 95% confidence intervals for the paired error reductions excluding zero. This supports choosing which expert transformations contribute to each input, making selective computation useful for forecasting accuracy as well as computational economy.

Semantic embeddings provide series meaning beyond numerical normalization. Their training-time sMAPE benefit has a 95% confidence interval excluding zero, while the MASE comparison narrowly misses the corresponding 5% threshold ($p = 0.0557$). Inference-only removal provides complementary evidence of learned reliance: withholding these cues increases FTS mixed MASE by 64.5% while keeping pretrained weights fixed.

6 Conclusion

FTFM combines Test-Time Learning Relational Attention, semantic conditioning, and shared forecasting layers for heterogeneous financial time series. With 27.42 million stored pretrained parameters, it achieves the lowest errors in most domain–horizon settings under each metric across eight domains, before and after LoRA adaptation. Its lead across Crypto, Equities, and Futures persists at every evaluated horizon after competing pretrained models receive the same benchmark training data.

Ablations support observed feature transitions as a useful learning signal beyond feature compatibility: removing prediction-error correction or replacing relational attention with ordinary attention increases average forecasting errors. Semantic embeddings complement normalization by preserving series meaning, with mixed-panel results highlighting their value for heterogeneous inputs. Selective expert routing also improves accuracy over activating all six experts, showing that choosing relevant transformations benefits prediction as well as limiting active computation.

These findings support a compact forecasting formulation that combines shared pretrained representations, supplied series semantics, and locally learned predictive relations. FTFM adapts cross-series information exchange to each observed history while keeping pretrained parameters fixed and requiring no future forecast labels.

AI USE STATEMENT

We used OpenAI Codex, with GPT-5.6 Sol and GPT-6 Astra, to assist with code development and with drafting and editing selected manuscript text for clarity, grammar, and readability. All AI-generated code was reviewed by the author. The author takes full responsibility for the final manuscript and associated research artifacts, including the implementation, experimental results, claims, and conclusions.

ETHICS STATEMENT

This study involved no human participants and used no identifiable customer, account, or transaction records. The paper evaluates financial time-series forecasting performance; live-trading evaluation is outside its scope. Improvements in forecasting accuracy do not establish trading profitability or the safety of live deployment, which also depend on execution conditions, transaction costs, and risk management. Forecast errors and changes in market conditions may cause financial losses if model outputs are used to guide financial decisions. Such applications require separate validation, ongoing monitoring, and appropriate human oversight. Commercially licensed pretraining data subject to redistribution restrictions will not be released, as described in the reproducibility statement. We have no relevant funding, vendor sponsorship, or commercial conflicts of interest to disclose.

REPRODUCIBILITY STATEMENT

All reproducibility materials and information described here are provided in the supplementary material and in a newly created anonymous GitHub repository at https://github.com/s-FQ/ICLR_27_FTFM_reproducibility. The repository contains no identifying information or links traceable to the authors or their affiliations. We provide the FTFM model implementation, pretrained and ablation checkpoints, trained LoRA adapters, and inference and evaluation code. For the baseline models, we provide the training and evaluation code, configurations, checkpoints, and trained LoRA adapters where applicable. We also provide the complete benchmark dataset, including its training, validation, and test splits, associated metadata, and forecast-origin definitions. Versioned dependencies, evaluation scripts, and scripts for generating the reported tables and figures accompany these materials. Experimental settings are described in Appendix D. We cannot redistribute the complete pretraining dataset because it contains data obtained from commercial providers, including Databento and Bloomberg, under licenses that prohibit redistribution. The provided materials support reproduction of the reported forecasting evaluations using the supplied checkpoints and adapters, but do not provide a complete reproduction of FTFM training.

REFERENCES

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. URL https://openreview.net/forum?id=gerNCVqqtR.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025. URL https://arxiv.org/abs/2510.15821.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. In *Advances in Neural Information Processing Systems*, volume 38, pp. 113506–113543. Curran Associates, Inc., 2025. doi: 10.52202/085713-3786. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a4ca07aa108036f80cbb5b82285fd4b1-Abstract-Conference.html.

Defu Cao, Furong Jia, Sercan Ö. Arık, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. TEMPO: Prompt-based generative pre-trained transformer for time series forecasting. In *Inter-*

national Conference on Learning Representations, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5132940b1bced8a7b28e9695d49d435a-Abstract-Conference.html.

Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/2be6705de7412adf107900add727a795-Abstract-Conference.html.

Si-An Chen, Chun-Liang Li, Nathanael C. Yoder, Sercan Ö. Arik, and Tomas Pfister. TSMixer: An all-MLP architecture for time series forecasting. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=wbpxTuXgm0.

Hanyin Cheng, Xingjian Wu, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. CoRA: Boosting time series foundation models for multivariate forecasting through correlation-aware adapter. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ae3e173398d6e43fea63cbcc16fbaa98-Abstract-Conference.html.

Ben Cohen, Emaad Khwaja, Youssef Doubli, Salahidine Lemaachi, Chris Lettieri, Charles Masson, Hugo Miccinilli, Elise Ramé, Qiqi Ren, Afshin Rostamizadeh, Jean du Terrail, Anna-Monica Toon, Kan Wang, Stephan Xie, Zongzhe Xu, Viktoriya Zhukova, David Asker, Ameet S. Talwalkar, and Othmane Abou-Amal. This time is different: An observability perspective on time series foundation models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 50907–50951, 2025. doi: 10.52202/085713-1698. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/48ea942b60e6c52b50ca90e4a40d8ac2-Abstract-Conference.html.

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. URL https://proceedings.mlr.press/v235/das24c.html.

Xueying Ding, Aakriti Mittal, and Achintya Gopal. DELPHYNE: A pre-trained model for general and financial time series. *arXiv preprint arXiv:2506.06288*, 2025. URL https://arxiv.org/abs/2506.06288.

Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam H. Nguyen, Wesley M. Gifford, Chandra Reddy, and Jayant Kalagnanam. Tiny time mixers (TTMs): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. In *Advances in Neural Information Processing Systems*, volume 37, pp. 74147–74181, 2024. doi: 10.52202/079017-2359. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/874a4d89f2d04b4bcf9a2c19545cf040-Abstract-Conference.html.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Hongbin Huang, Minghua Chen, and Xiao Qiao. Generative learning for financial time series with irregular and scale-invariant patterns. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/f90fc76b199fe6b0ec2a51aaf72c3277-Abstract-Conference.html.

Ayush Jain and Rajat Sen. TimesFM-3: A zero-shot foundation model for multivariate forecasting. Google Research Blog, August 2026. URL https://www.research.google/blog/timesfm-3-a-zero-shot-foundation-model-for-multivariate-forecasting/.

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. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/680b2a8135b9c71278a09cafb605869e-Abstract-Conference.html.

Emaad Khwaja, Chris Lettieri, Gerald Woo, Eden Belouadah, Marc Cenac, Guillaume Jarry, Enguerrand Paquin, Xunyi Zhao, Viktoriya Zhukov, Othmane Abou-Amal, Chenghao Liu, Ameet Talwalkar, and David Asker. Toto 2.0: Time series forecasting enters the scaling era. *arXiv preprint arXiv:2605.20119*, 2026. URL https://arxiv.org/abs/2605.20119.

Kelvin J. L. Koa, Jan Chen, Yunshan Ma, Huanhuan Zheng, and Tat-Seng Chua. Reasoning on time-series for financial technical analysis. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/68c1b9784d982fa5575bb45ace7bd64e-Abstract-Conference.html.

Hunjae Lee and Corey Clark. Dynamic relational priming improves transformer in multivariate time series. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2509.12196v2.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qrwe7XHTmYb.

Junjie Li, Yang Liu, Weiqing Liu, Shikai Fang, Lewen Wang, Chang Xu, and Jiang Bian. MarS: A financial market simulation engine powered by generative foundation model. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/6261d3bc9e9326ddff055595aabd54e1-Abstract-Conference.html.

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*, volume 38, pp. 162–170, 2024. doi: 10.1609/aaai.v38i1.27767. URL https://ojs.aaai.org/index.php/AAAI/article/view/27767.

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. doi: 10.1016/j.ijforecast.2021.03.012. URL https://arxiv.org/abs/1912.09363.

Chenghao Liu, Taha Aksu, Juncheng Liu, Xu Liu, Hanshu Yan, Quang Pham, Silvio Savarese, Doyen Sahoo, Caiming Xiong, and Junnan Li. Moirai 2.0: When less is more for time series forecasting. *arXiv preprint arXiv:2511.11698*, 2025a. URL https://arxiv.org/abs/2511.11698.

Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Junnan Li, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Moirai-MoE: Empowering time series foundation models with sparse mixture of experts. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 38940–38962, 2025b. URL https://proceedings.mlr.press/v267/liu25an.html.

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

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*, 2025c. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d09820e292f915d77fa717d228c425ee-Abstract-Conference.html.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The M4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1): 54–74, 2020. doi: 10.1016/j.ijforecast.2019.04.014.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9355–9366, 2021. URL https://proceedings.mlr.press/v139/schlag21a.html.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-MoE: Billion-scale time series foundation models with mixture of experts. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/558d48c1f08675daa636e09bfe94a89e-Abstract-Conference.html.

Yu Shi, Zongliang Fu, Shuo Chen, Bohan Zhao, Wei Xu, Changshui Zhang, and Jian Li. Kronos: A foundation model for the language of financial markets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25366–25373, 2026. doi: 10.1609/aaai.v40i30.39730. URL https://ojs.aaai.org/index.php/AAAI/article/view/39730.

Yejun Soun, Hosung Lee, Suyoung Park, and U Kang. STABLE: Shift-tolerant allocation via Black–Litterman using conditional diffusion estimates. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/4d4dbc5c955c5d273eed12d565335068-Abstract-Conference.html.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (Learn at test time): RNNs with expressive hidden states. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57503–57522. PMLR, 2025. URL https://proceedings.mlr.press/v267/sun25h.html.

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*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/a7ac8a21e5a27e7ab31a5f42a0117bdb-Abstract-Conference.html.

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*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/2b187165e28fdfdc0ffb34d1bfff2b0c-Abstract-Conference.html.

Aaron Wheeler and Jeffrey D. Varner. MarketGPT: Developing a pre-trained transformer (GPT) for modeling financial time series. *arXiv preprint arXiv:2411.16585*, 2024. URL https://arxiv.org/abs/2411.16585.

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. URL https://proceedings.mlr.press/v235/woo24a.html.

Wentao Xu, Weiqing Liu, Lewen Wang, Yingce Xia, Jiang Bian, Jian Yin, and Tie-Yan Liu. HIST: A graph-based framework for stock trend forecasting via mining concept-oriented shared information. *arXiv preprint arXiv:2110.13716*, 2021. URL https://arxiv.org/abs/2110.13716.

Yuanjian Xu, Jianing Hao, Anxian Liu, Zhenzhuo Li, Shichang Meng, Shuai Yuan, and Guang Zhang. LENS: Large pre-trained transformer for exploring financial time series regularities. In *Proceedings of the 6th ACM International Conference on AI in Finance*, pp. 771–778, 2025. doi: 10.1145/3768292.3770349. URL https://doi.org/10.1145/3768292.3770349.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In *Advances in Neural Information Processing Systems*, volume 37, pp. 115491–115522, 2024. doi: 10.52202/079017-3668. URL

https://proceedings.neurips.cc/paper_files/paper/2024/hash/d13a3eae72366e61dfdc7eea82eeb685-Abstract-Conference.html.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/4904fad153f6434a7bcf04465d4be2cc-Abstract-Conference.html.

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. doi: 10.1609/aaai.v37i9.26317. URL https://ojs.aaai.org/index.php/AAAI/article/view/26317.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vSVLM2j9eie.

Lifan Zhao and Yanyan Shen. Rethinking channel dependence for multivariate time series forecasting: Learning from leading indicators. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JiTVtCUOpS.

Wanyun Zhou, Saizhuo Wang, Mihai Cucuringu, Zihao Zhang, Xiang Li, Jian Guo, Chao Zhang, and Xiaowen Chu. DeltaLag: Learning dynamic lead-lag patterns in financial markets. In *Proceedings of the 6th ACM International Conference on AI in Finance*, pp. 422–430, 2025. doi: 10.1145/3768292.3770421. URL https://arxiv.org/abs/2511.00390.

Zhuohang Zhu, Haodong Chen, Qiang Qu, and Vera Chung. FinCast: A foundation model for financial time-series forecasting. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 4539–4549, 2025. doi: 10.1145/3746252.3761261. URL https://arxiv.org/abs/2508.19609.

A Extended Results and Ablations

A.1 Qualitative forecast comparisons

Figures 2–5 show the six pretrained models in Table 1. Each comparison uses the same series and forecast origin across models, the domain-specific input context, and the longest horizon reported for that domain. Blue lines show the observed history and realized future in original units; red lines show the forecasts, with a dashed vertical line marking the forecast origin. Axes are shared across models within each figure. These selected examples illustrate forecast behavior; aggregate accuracy is reported in Table 1.

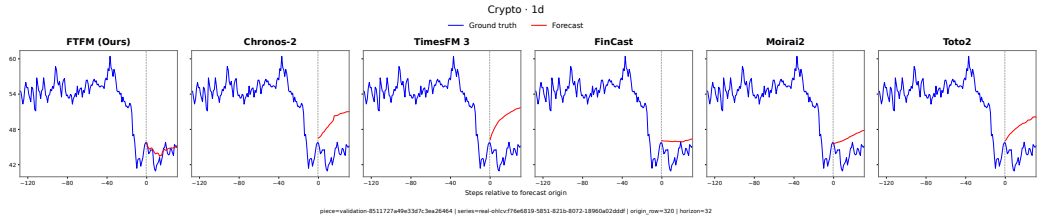


Figure 2: Qualitative forecasts for a daily Crypto series with 128 context steps and a 32-step forecast horizon.

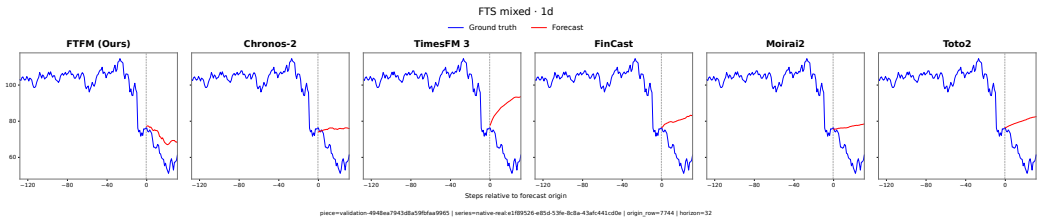


Figure 3: Qualitative forecasts for a daily FTS mixed series with 128 context steps and a 32-step forecast horizon.

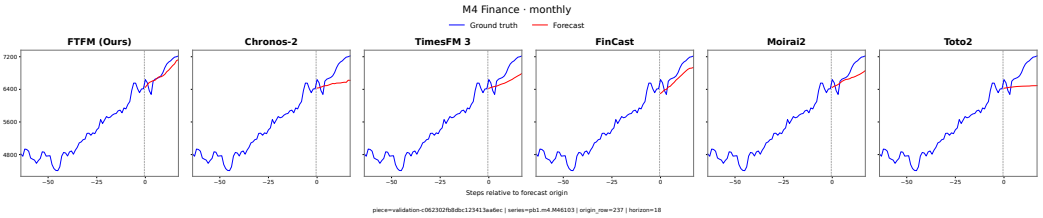


Figure 4: Qualitative forecasts for a monthly M4 Finance series with 64 context steps and its native 18-step forecast horizon.

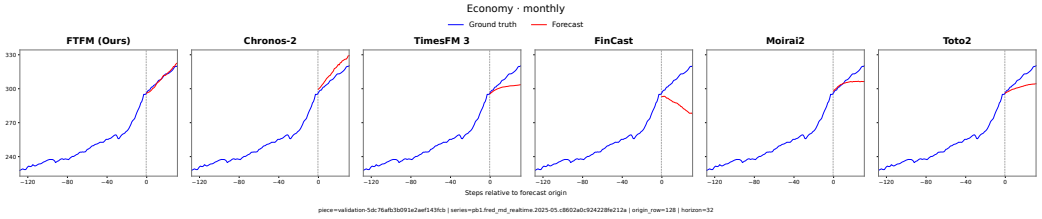


Figure 5: Qualitative forecasts for a monthly Economy series with 128 context steps and a 32-step forecast horizon.

B Mechanism Analysis and Case Studies

B.1 Relation-Memory Updates and Attention

We trace a real CME futures panel from the Futures benchmark, retaining all five series: Open, High, Low, Close, and Volume. The 32 observed positions span 19:30:05–19:30:45 UTC on 21 May 2026 (example `18061f549541ee4a3129bb48`, origin row 27740). They form eight four-position patches. The source retains observed bars with uneven timestamp gaps; we preserve the benchmark's nominal one-second cadence and do not insert observations.

For a readable case study, we inspected the central test origin in each of 72 native five-series Futures pieces and retained 25 windows with High $>$ Low in at least eight bars. Across all 16 blocks and eight heads, we ranked traces by off-diagonal attention modulation weighted by relative memory change. Selection used observed contexts and internal diagnostics, without future values or forecast errors; the selected case does not characterize typical effect sizes. Figure 6 shows the zero initialization and seven observed-transition updates, with both attention paths computed from the same current temporal features.

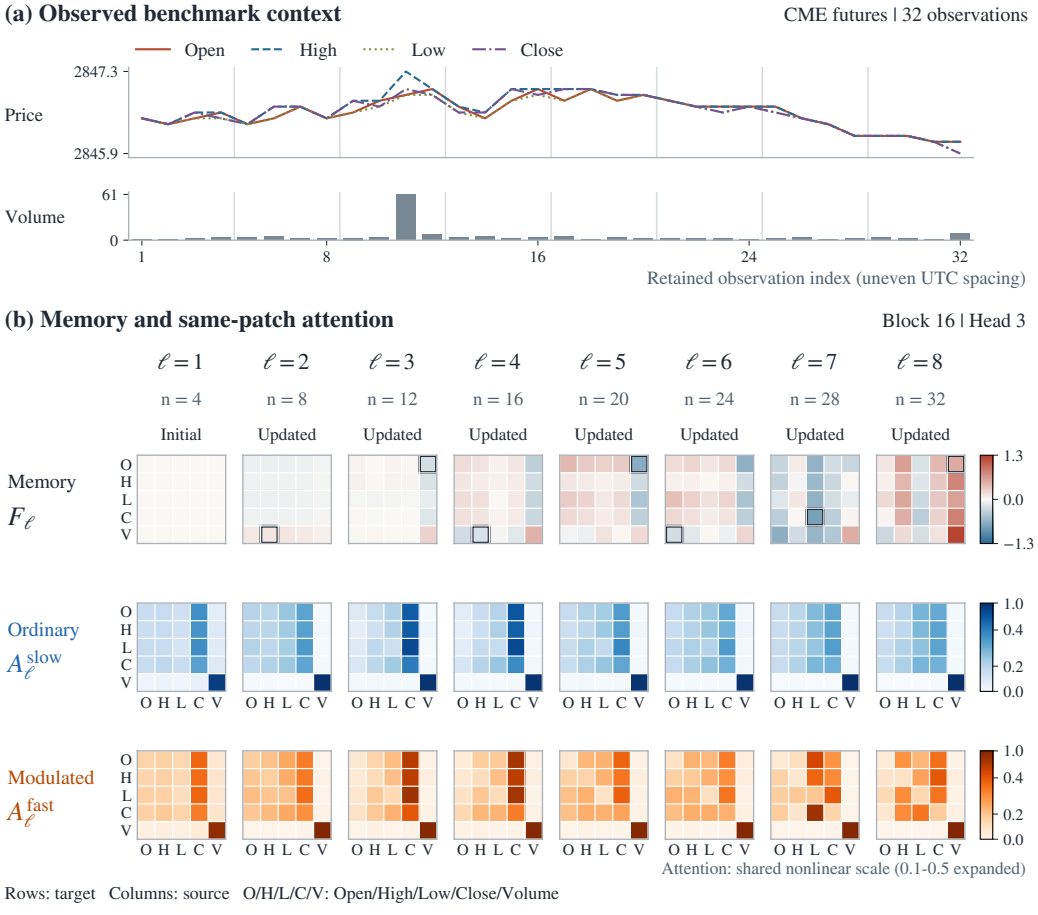


Figure 6: A real benchmark context and the resulting directed predictive relation memory. The top panel shows original OHLCV values. Each heatmap column uses the observed prefix ending at position $n = 4\ell$. Rows are targets and columns are sources; outlines mark the largest off-diagonal correction at each update. All memory snapshots share one signed scale. Ordinary (blue) and relation-modulated (orange) attention share a nonlinear probability scale expanded between 0.1 and 0.5; the full range from 0 to 1 is retained.

Figure 7 expands patch 7, which contains the largest off-diagonal correction after the initial learned state. For the Low-to-Close coefficient, the retained value $\bar{F}_{7,CL} \simeq 0.090$ receives a correction of approximately -0.891, yielding $F_{7,CL} \simeq -0.800$. The next observed transition revises this coefficient to approximately -0.392. Thus the memory revises both the sign and magnitude of its local predictive coefficients. These coefficients fit learned temporal probes, not raw-price correlations or causal effects. For this head, the learned update rate is $\eta \simeq 0.9947$ and the per-patch retention is $\exp(-4\rho) \simeq 0.999394$; the displayed revision is dominated by the prediction-error correction.

The lower panel follows the same state into value aggregation. The negative coefficient gives a positive strength $R_{7,CL} = |\tanh(F_{7,CL})| \simeq 0.664$. With learned $\beta \simeq 1.6839$, the corresponding attention probability changes from $A^{\text{slow}}_{7,CL} \simeq 0.285$ to $A^{\text{fast}}_{7,CL} \simeq 0.500$. The final value coefficient is $W_{7,CL} \simeq 0.288$, because the learned blend weight is only $\lambda \simeq 0.0454$:

$$W_7 = (1-\lambda)A_7^{\text{slow}} + \lambda\big(A_7^{\text{fast}} \odot R_7\big), \qquad O_7 = W_7 V_7. \tag{7}$$

The effective value weights are not renormalized. The trace illustrates context-local revision and its bounded contribution to value aggregation; it does not establish a forecasting gain.

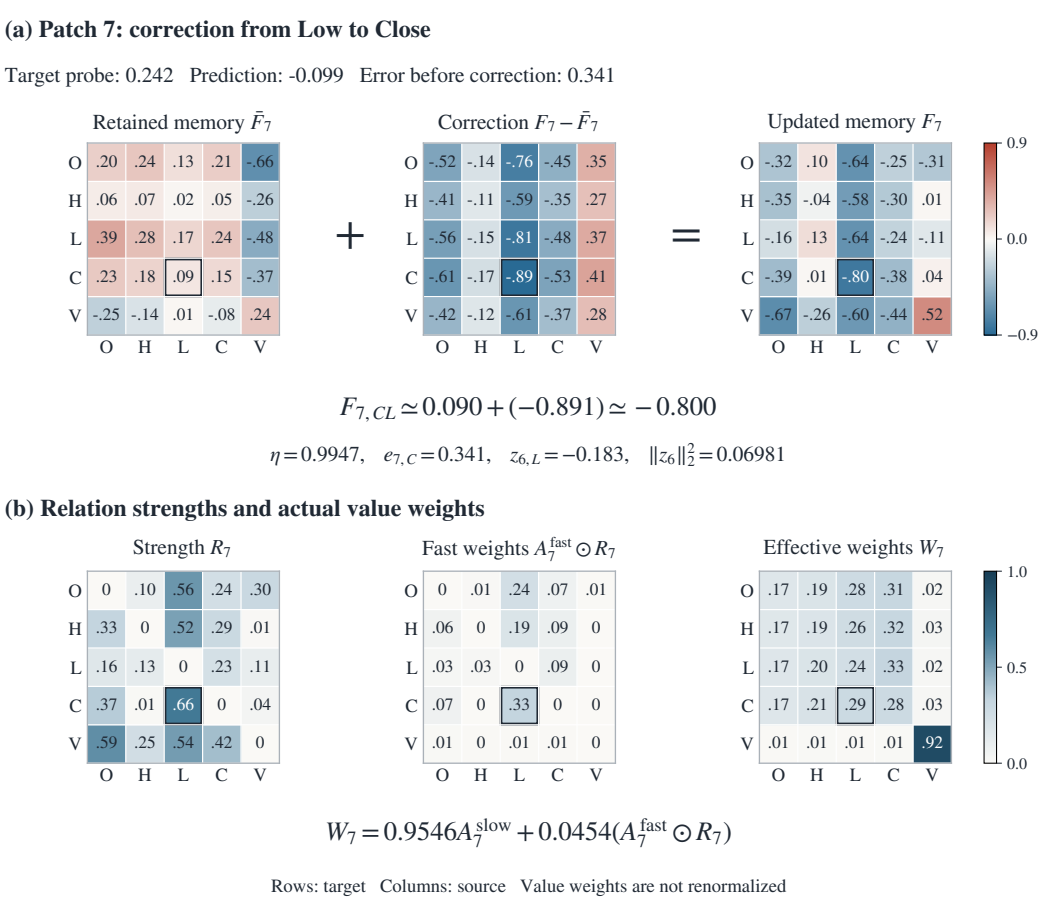


Figure 7: Measured prediction-error correction and value aggregation at patch 7. The upper row decomposes the updated memory into retained state and residual correction; the lower row shows relation strengths, fast-path value weights, and effective value weights. The outlined cell is Close←Low. Values are rounded; color scales are fixed within each row. The signed memory includes self-lags, whereas R_7 removes them.

B.2 Expert Routing

Figure 8 traces the frozen checkpoint in CPU FP32 on a 32-observation Futures OHLCV context (example `b98f6ee32893fd090d06c45f`). Each four-observation patch produces five tokens. The router selects two of six experts per token and normalizes their gates over that pair.

We screened the central H4 test origin in each of 72 Futures pieces, retaining 25 complete contexts with nonzero high–low ranges in at least eight bars. We selected the context and block maximizing the average of temporal changes and cross-series differences in the selected expert pairs, measured by one minus half their intersection size. Selection uses observed contexts and routing only, without future values or forecast errors.

For clarity, we show only Close and Volume in block 4. Close switches from experts 1/4 at patch 1 to 1/2 at patch 2; Volume selects 5/3 at patch 1. All six experts participate across the context, with two active per token. The all-expert ablation in Section 5 separately assesses the forecasting benefit.

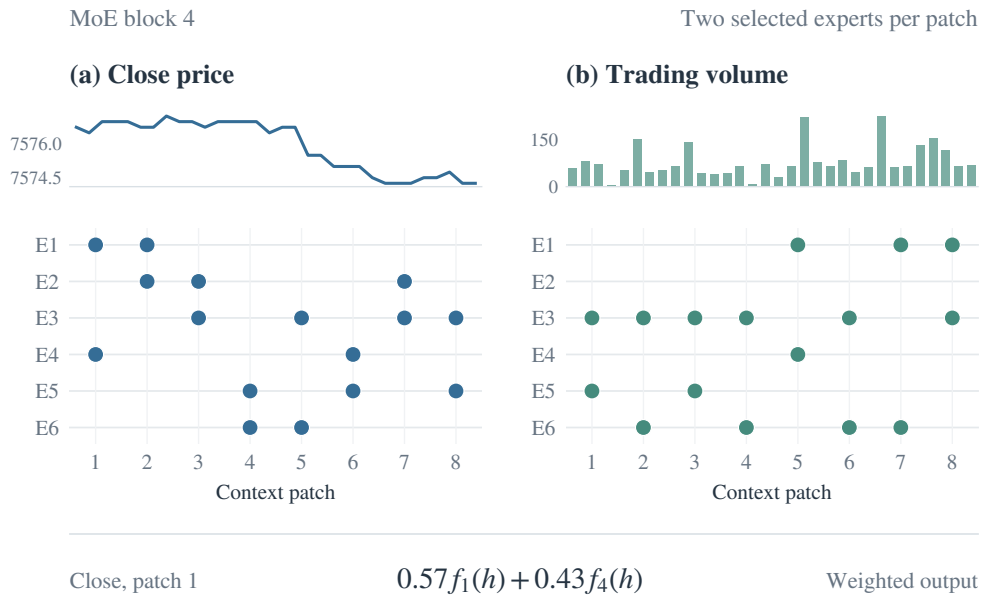


Figure 8: Expert choices for Close and Volume in MoE block 4. The observed histories above each panel align with the eight four-observation patches below. Each column has two dots marking the selected experts; dot size does not encode gate weight. The bottom example shows one token's weighted expert sum, with rounded gates. Expert IDs are local to the block. This selected case does not characterize typical usage.

C METHOD DETAILS

The optional notation macros bundled with the ICLR template are adapted from Goodfellow et al. (2016). This appendix expands the representations, update rules, and training objective in Section 3. Symbols for patch length, forecast horizon, quantile grid, and loss coefficients refer to the fixed settings below.

C.1 INPUT REPRESENTATION AND NORMALIZATION

The model uses non-overlapping patches of $P = 4$ steps. For a real series s, define the observed, time-valid prefix positions and their count by

$$\mathcal{O}_{\ell s} = \{t \le \ell P : M_{ts} = 1,\ \tau_t = 1\}, \qquad n_{\ell s} = |\mathcal{O}_{\ell s}|. \tag{8}$$

With a fixed scale floor $\sigma_{\min} > 0$, the prefix statistics are

$$\mu_{\ell s} = \begin{cases} n_{\ell s}^{-1} \sum_{t \in \mathcal{O}_{\ell s}} x_{ts}, & n_{\ell s} > 0, \\ 0, & n_{\ell s} = 0, \end{cases}$$
$$\sigma_{\ell s} = \begin{cases} \max\left\{ \sigma_{\min}, \sqrt{\dfrac{\sum_{t \in \mathcal{O}_{\ell s}} (x_{ts} - \mu_{\ell s})^2}{n_{\ell s} - 1}} \right\}, & n_{\ell s} \ge 2, \\ \sigma_{\min}, & n_{\ell s} < 2. \end{cases} \tag{9}$$

These are the masked cumulative mean and floored sample standard deviation. An empty real prefix uses zero location and the scale floor; the formulas above exclude wholly padded series. Each patch retains its own statistics, detached from gradients. Statistics include observations through the patch endpoint, so causality is defined at patch boundaries: later patches cannot change earlier tokens. No statistics are fitted to a separate dataset.

Equation 1 applies the asinh transform to observed values. Missing and padded entries receive transformed value zero, with M and τ retained in $\phi_{\ell s}$ to distinguish them from observed zeros. The three P-length vectors give $\phi_{\ell s} \in \mathbb{R}^{3P}$. Numerically missing observations and structural padding remain distinct: missing values within a real patch do not by themselves invalidate its token.

The shorthand maps in Equation 2 expand to

$$h_{\ell s}^{(0)} = W_p \phi_{\ell s} + b_p + w_\Delta \frac{\log(\Delta_{\text{nom}}/\Delta_0)}{c_\Delta} + \gamma W_m \frac{E_q(q_s) + E_f(f_s) + E_r(r_s)}{\sqrt{3}}. \tag{10}$$

Here E_{patch} is the shared affine patch projection, E_{cad} is the logarithmic cadence term, and E_{sem} is the semantic term. The low-rank tables E_q, E_f, E_r encode quantity family, field role, and representation; their sum is scaled by $1/\sqrt{3}$ and mapped through W_m to hidden width D, with a bounded learned scale γ. The constants Δ_0 and c_Δ are fixed. Cadence specifies nominal step duration; calendar phase and realized gaps are not inputs.

During training, randomly selected metadata fields or complete descriptors are replaced with unknown categories. This preserves a shared representation for missing descriptors without adding instrument-identity or series-position embeddings. Supplied descriptors identify the meaning of a series: for example, a volatility descriptor identifies the quantity being forecast rather than estimating current volatility from the observations.

C.2 BACKBONE CONFIGURATION AND MASKING

Each decoder block applies pre-normalized residual sublayers for causal temporal attention, Test-Time Learning Relational Attention, and feed-forward processing. Temporal attention uses rotary position embeddings on queries and keys and attends only to patches $m \le \ell$ along the same series. Dense SwiGLU and mixture-of-experts sublayers alternate; expert sublayers use top-two routing (Shazeer et al., 2017; Lepikhin et al., 2021), and a final RMSNorm precedes the forecast heads. Model dimensions, expert counts, and optimization settings are listed in Appendix D.

Wholly padded patches and padded series are excluded from attention, and their states are zeroed after each residual branch. Numerical missingness instead remains represented by the input masks.

Shared projections and feed-forward parameters preserve equivariance to joint permutations of series, masks, and metadata at inference, while stored parameter count is independent of panel width.

Within the relational attention sublayer, the probe and attention projections use the same normalized post-temporal features. Separate projections produce the scalar probes z_ℓ and the attention quantities Q_ℓ, K_ℓ, V_ℓ; the latter are shared between ordinary and relation-modulated attention. At deeper blocks, temporal features also contain information exchanged by earlier cross-series sublayers. Thus the lag-one task in Equation 3 predicts transitions between history-conditioned representations rather than adjacent raw values.

C.3 Relation-memory learning and attention

Each sample, block, and head initializes an independent $F_1 = 0$ and scans patches in order. The memory is separate from pretrained parameters θ and is reconstructed on each model call. Its entry $F_{\ell,sr}$ predicts target probe s from source probe r. The local predictor includes diagonal self-lags; the attention strength matrix removes the diagonal through $R_{\ell,sr} = \mathbf{1}[s \neq r]|\tanh(F_{\ell,sr})|$. Equal-magnitude positive and negative coefficients therefore produce the same relation strength. Self-lags contribute to transition prediction, while the zero diagonal of R_ℓ excludes self-values from relation-modulated aggregation.

For $\ell \geq 2$, let $\mathcal{T}_\ell = \mathcal{P}_{\ell-1} \cup \mathcal{P}_\ell$ be the two patches forming a transition. With s ranging over real series, the update gate is

$$g_\ell = \mathbf{1}[M_{ts} = 1 \text{ and } \tau_t = 1 \quad \text{for every } s \in \{1, \ldots, S\},\ t \in \mathcal{T}_\ell]\,. \tag{11}$$

Both endpoint patches must therefore be completely observed and time-valid across the entire real panel. Incomplete endpoint patches skip the prediction-error correction. A partially observed patch may still participate in forecasting, and cross-series attention can use the decayed relation memory. When $g_\ell = 0$, Equation 3 gives $F_\ell = \bar{F}_\ell$: the memory decays rather than freezing or resetting. The duration $\Delta_\ell = P\Delta_{\text{nom}}$ uses nominal time, so scheduled closures contribute no additional elapsed time to the decay.

For fixed probes, differentiating the local objective $\mathcal{J}_\ell$ defined in Section 3.4 gives

$$\begin{aligned} \nabla_F \mathcal{J}_\ell(F) &= (F z_{\ell-1} - z_\ell) z_{\ell-1}^\top, \\ \nabla_F \mathcal{J}_\ell(\bar{F}_\ell) &= -e_\ell z_{\ell-1}^\top. \end{aligned} \tag{12}$$

The corresponding normalized gradient step after decay is

$$F_\ell = \bar{F}_\ell - \frac{g_\ell \eta}{\epsilon + \|z_{\ell-1}\|_2^2} \nabla_F \mathcal{J}_\ell(\bar{F}_\ell). \tag{13}$$

Substituting Equation 12 into Equation 13 recovers the normalized residual correction in Equation 3. The denominator uses the energy of all preceding probes. Holding probes fixed applies only to this derivative with respect to F: forecasting gradients still pass through the probes, temporal attention, memory recurrence, and cross-series attention during pretraining. The local loss defines the update and is not added to the forecasting objective.

For each transition, the computation proceeds as follows:

1. Decay the preceding state to obtain $\bar{F}_\ell$ and predict the current probes as $\bar{F}_\ell z_{\ell-1}$.
2. Form $e_\ell = z_\ell - \bar{F}_\ell z_{\ell-1}$. Apply the normalized correction only if $g_\ell = 1$; otherwise retain the decayed state.
3. Construct R_ℓ from the updated F_ℓ, and compute both attention maps using the shared current queries Q_ℓ and keys K_ℓ.
4. Combine the paths using Equation 5. Concatenate head outputs and project them before residual addition.

The updated state is used at the same patch, whose probes are already observed within the prefix. The first patch uses the initialized F_1. Both paths remain dependent on current temporal features; "slow" and "fast" distinguish ordinary attention from its relation modulation.

Each block and head learns $0 < \eta < 1$ and $\rho > 0$, with $\epsilon > 0$ stabilizing the correction. The attention scales use capped sigmoids,

$$\beta = 2\,\mathrm{sigmoid}(b_\beta), \qquad \lambda = \tfrac{1}{4}\,\mathrm{sigmoid}(b_\lambda), \tag{14}$$

where b_β, b_λ are learned scalars, giving $0 < \beta < 2$ and $0 < \lambda < 0.25$. For a real query, $\lambda_{\text{eff}} = \lambda$ when another real series exists, and zero otherwise. This structural condition is distinct from the observation gate: a transition with $g_\ell = 0$ can still use the decayed relation memory in attention.

Both softmaxes include the valid self key and use the same source mask. The self key therefore still participates in softmax normalization, even though $R_{\ell,ss} = 0$ removes its value contribution from the relation-modulated path. The multiplication $A^{\text{fast}}_\ell \odot R_\ell$ is not renormalized. Its row sum, $\sum_r A^{\text{fast}}_{\ell,sr} R_{\ell,sr}$, lies between zero and one. When relation strengths are small, the total aggregation weight remains small; the weights are not rescaled to sum to one. Consequently, with multiple real series and $F_\ell = 0$,

$$O_\ell = (1 - \lambda) A^{\text{slow}}_\ell V_\ell. \tag{15}$$

A singleton panel instead has $\lambda_{\text{eff}} = 0$. The sign of F matters to local prediction, whereas only its bounded magnitude guides attention. These coefficients are predictive associations in the learned probe space, not causal effects or calibrated measures of forecasting usefulness.

C.4 Forecast heads and objective settings

The fixed patch length is $P = 4$, the direct quantile horizon is $H_{\max} = 16$, and the eleven quantile levels are $\mathcal{Q} = \{0.01, 0.05, 0.10, 0.25, 0.40, 0.50, 0.60, 0.75, 0.90, 0.95, 0.99\}$. A shared linear head predicts the next four values per series. The separate quantile head uses learned horizon embeddings and positive softplus increments to produce ordered marginal quantiles over sixteen steps. These head dimensions are fixed; longer benchmark horizons use the recursive protocol in Appendix D.

Training decodes eligible prefixes from $h_2, \ldots, h_L$ in one causal backbone pass. A prefix belongs to $\mathcal{E}$ if its final patch has no time padding, it contains at least two such patches, and each head has at least one available target. This forecasting eligibility differs from the complete-observation gate for relation updates. The overlapping sixteen-step targets supervise multiple causal origins; the four-step point head emphasizes the next patch.

For prefix i ending at patch ℓ_i, let $y_{i,sk}$ be a future target for series s at lead k. Suppressing the sample index on the normalization statistics, every head uses origin-owned targets

$$u_{i,sk} = \operatorname{asinh}\left(\frac{y_{i,sk} - \mu_{\ell_i s}}{\sigma_{\ell_i s}}\right). \tag{16}$$

The statistics come from the forecast origin, not from future observations. Let $\mathcal{V}_{\mathrm{p},i}$ and $\mathcal{V}_{Q,i}$ be the available series–lead target cells for the point and quantile heads. Their masked means are

$$\begin{aligned} \mathcal{L}_{\mathrm{p},i} &= \frac{1}{|\mathcal{V}_{\mathrm{p},i}|} \sum_{(s,k)\in\mathcal{V}_{\mathrm{p},i}} \mathrm{Huber}_1(u_{i,sk} - \hat{u}_{\mathrm{p},i,sk}), \\ \mathcal{L}_{Q,i} &= \frac{1}{|\mathcal{V}_{Q,i}|\,|\mathcal{Q}|} \sum_{(s,k)\in\mathcal{V}_{Q,i}} \sum_{q\in\mathcal{Q}} \ell_q(u_{i,sk} - \hat{u}_{Q,i,sk}(q)), \end{aligned} \tag{17}$$

where $\hat{u}_{\mathrm{p}}$ and $\hat{u}_Q(q)$ denote the normalized predictions, and

$$\mathrm{Huber}_1(r) = \begin{cases} \frac{1}{2}r^2, & |r| \le 1, \\ |r| - \frac{1}{2}, & |r| > 1, \end{cases} \qquad \ell_q(r) = (q - \mathbf{1}[r < 0])r. \tag{18}$$

Eligibility ensures a nonzero target count for each head. The point-head loss is also denoted $\mathcal{L}_{4,i}$ because $P = 4$; thus $\mathcal{L}_{\mathrm{p},i} = \mathcal{L}_{4,i}$ in the evaluated model. The fixed coefficients in Equation 6 are $\alpha_{\mathrm{p}} = \alpha_{\mathrm{q}} = 1/2$, $\alpha_{\mathrm{b}} = 0.01$, and $\alpha_{\mathrm{z}} = 0.001$, giving

$$\mathcal{L} = \frac{1}{|\mathcal{E}|} \sum_{i\in\mathcal{E}} \left(\tfrac{1}{2}\mathcal{L}_{4,i} + \tfrac{1}{2}\mathcal{L}_{Q,i}\right) + 0.01\mathcal{L}_{\text{balance}} + 0.001\mathcal{L}_z. \tag{19}$$

Each eligible prefix has equal outer weight regardless of its panel width or number of available target cells. The router terms instead use token weighting: $\mathcal{L}_{\text{balance}}$ encourages balanced expert use, and $\mathcal{L}_z$ penalizes squared log-sum-exp routing logits. Neither term is the local transition loss $\mathcal{J}_\ell$.

Inference decodes the final context state. Predictions are restored to raw units by $\hat{x} = \mu_{Ls} + \sigma_{Ls} \sinh(\hat{u})$. The $q = 0.50$ output provides median forecasts for $1 \leq H \leq 16$; the Huber-trained four-step point forecast remains a separate output. The quantiles are marginal over future times and series and do not define a joint predictive distribution.

D Experimental Details

D.1 Benchmark composition and splits

Table 5 summarizes the economic coverage, forecast variables, and panel construction of each benchmark group. Panel width counts concurrent scalar series, including separate financial fields; it is not necessarily the number of instruments. The domain, cadence, and width views classify the same underlying pieces and forecast origins. Each model/configuration generates one set of forecasts for all three views; the views are not independent datasets and their aggregate scores are not averaged together.

Table 5: Data coverage and forecasting panels in the evaluation benchmark. OHLCV denotes open, high, low, and close prices and trading volume. Cadence is the nominal observation interval; panel width S counts concurrent scalar series, including separate fields of the same instrument.

Benchmark group	Data coverage	Variables / panel construction	Cadence	S
Crypto	Cryptocurrency spot markets	OHLCV fields from one or multiple instruments; selected-field panels	1 second, 1 minute, daily, weekly	1–44
Equities	US-listed stocks and exchange-traded products	One instrument's five OHLCV fields	1 second, 1 minute	5
Futures	Exchange-traded futures across financial and commodity markets	One futures instrument's five OHLCV fields	1 second	5
Exchange	Eight foreign-exchange rate series	Individual exchange-rate levels; one series per panel	Daily	1
M4 Finance	Historical financial series in M4's Finance category	Individual series in source units	Daily, weekly, monthly, quarterly, yearly	1
Economy	US activity, employment, prices, money, credit, and financial indicators (FRED-MD)	Individual indicator levels across historical and real-time vintages	Monthly	1
	US macroeconomic and financial indicators, including balance-sheet measures (FRED-QD)	Individual indicator levels across real-time vintages	Quarterly	1
	Historical macroeconomic series in M4's Macro category	Individual series in source units	Daily, weekly, monthly, quarterly, yearly	1
FTS mixed	International stocks, funds, indices, currencies, futures proxies, and cryptocurrencies	Instrument-level OHLCV and panels selecting fields across instruments	1 minute, 1 hour, daily	1–50
Synthetic	Simulated financial quantities with controlled dependencies and control cases	Log returns, log volume, log volatility, and rate levels arranged into panels	1 hour	1–16

Sources: Binance (Crypto); Databento US securities and CME futures (Equities and Futures); the Exchange-rate benchmark (Exchange); M4 (Finance and Macro); FRED-MD and FRED-QD (US indicators); FTS (mixed markets); and generated data (Synthetic). M4 metadata supplies category labels but does not identify individual instruments or detailed variable semantics. Synthetic panels include autoregressive, delayed-factor, nonlinear, and cointegrating processes, together with null and other control panels.

During market-data acquisition, duplicate coverage is reviewed using canonical instrument identity, observation representation, cadence, and overlapping calendar intervals, including vendor aliases. These checks address repeated acquisition of the same coverage.

For non-M4 data, the chronological 60/20/20 boundaries apply within each source trajectory's selected span. Forecast horizons are 4, 16, and 32 sampling steps. Origins advance by the forecast horizon, and the entire target block must lie inside its assigned split and piece. Overlapping pieces are deduplicated by trajectory, variant, series identities, origin, and horizon, retaining the piece with the longer available history. For M4, the final native horizon of the official training history is reserved for validation, rolling training origins precede this segment, and the official held-out horizon supplies the test target. Native M4 horizons are 6, 8, 13, 14, and 18 for yearly, quarterly, weekly, daily, and monthly series, respectively. In Economy, horizons 4, 16, and 32 refer to FRED, while the native M4 horizons refer to M4 Macro.

Tables 1 and 2 report 26 of the 31 domain–horizon settings in the benchmark grid. To keep the main comparison concise, we omit weekly M4 Finance ($h = 13$) and yearly, quarterly, weekly, and monthly M4 Macro ($h = 6, 8, 13, 18$, respectively). These forecast step counts coincide with or are close to horizons already displayed: $h = 6, 8, 18$ appear under M4 Finance, while $h = 13$ is adjacent to the reported $h = 14$. This reporting choice reduces repetition in horizon lengths, although the underlying subsets differ in sampling frequency and series composition. First-place counts and majority statements refer specifically to the 26 reported settings.

A forecast never receives its future targets or their availability masks.

D.2 FTFM INFERENCE

Both base and adapted evaluations use the domain context policy in Section 4. For horizons above sixteen, both evaluations append a predicted median block, keep the latest domain-specific number of context steps, and rerun the complete model. Normalization and relation states are reconstructed on each call; generated values participate in subsequent conditioning without access to future truth.

During recursive rollout, generated values are marked available and participate in relation-memory updates when the endpoint coverage requirements are met. For transitions whose target patch is generated, the local residual compares predicted probes with features derived from model-generated values, rather than newly observed outcomes. These corrections therefore do not measure forecast error against future ground truth.

D.3 METRIC DEFINITIONS AND AGGREGATION

For each piece, scalar series, and native horizon, let $\mathcal{V}$ contain the valid origin–lead pairs. With lag-one scale a and $a^\star = 1$ when $a \le 10^{-8}$ (otherwise $a^\star = a$), we compute

$$\mathrm{MASE} = \frac{1}{|\mathcal{V}|} \sum_{(o,k)\in\mathcal{V}} \frac{|y_{o,k} - \hat{y}_{o,k}|}{a^\star}, \tag{20}$$

$$\mathrm{sMAPE} = \frac{1}{|\mathcal{V}|} \sum_{(o,k)\in\mathcal{V}} \frac{200|y_{o,k} - \hat{y}_{o,k}|}{|y_{o,k}| + |\hat{y}_{o,k}|}, \tag{21}$$

where a both-zero sMAPE summand is defined as zero. The scale a is the mean absolute difference over finite adjacent observations before the supplied scale boundary: the end of validation for non-M4 test data and the end of the official training history for M4 test data. Scale history is independent of the requested input context. If the scale history contains no valid adjacent pair, MASE is undefined; those targets remain in sMAPE. We use this common lag-one convention for all domains, including M4, rather than M4's seasonal scale or OWA.

Within a domain–horizon row, piece–series scores receive equal weight. When reporting a domain average, horizons receive equal weight; a view-level average then weights categories equally. These are not global averages over all scalar targets or averages over unique instruments across pieces. The foundation-model exports retain the full horizon grid. Each displayed comparison uses the corresponding domain–horizon subset.

D.4 FTFM PRETRAINING DETAILS

Section 4.1 describes the corpus and core optimization settings. The model has 16 blocks, hidden width 256, eight attention heads, and four-step patches; every second block uses six experts with top-two routing. The master seed is 2027.

AdamW (Loshchilov & Hutter, 2019) uses $(\beta_1, \beta_2) = (0.9, 0.95)$ and $\epsilon = 10^{-8}$. The learning-rate schedule spans 760,407 updates, with 5% linear warmup, 80% constant learning rate, and 15% geometric decay from a peak of 2×10^{-4} to 2×10^{-5}.

D.5 ADDITIONAL FINANCIAL FOUNDATION MODELS

This subsection supplements the model selection described in Section 4. Our pretrained comparisons require publicly accessible checkpoints and inference implementations that can produce forecasts from the observations supplied by the benchmark. The following financial models are omitted because we could not verify the necessary public releases or because their input requirements differ from our evaluation setting.

DELPHYNE. DELPHYNE (Ding et al., 2025) addresses both general and financial time series, including multivariate conditioning and missing observations, making it relevant to our forecasting setting. However, our search of the paper, author pages, and public repositories on 26 September 2026 did not identify an official downloadable pretrained checkpoint and corresponding inference implementation. We therefore do not include DELPHYNE in the empirical comparison.

LENS. LENS (Xu et al., 2025), introduced in an earlier manuscript version as PLUTUS, is pretrained on financial time series and models temporal and cross-channel dependencies. Although the initial PLUTUS manuscript describes the model as open-source, our search on 26 September 2026 did not identify an official public release of pretrained weights and inference code. We could therefore neither verify that release claim nor evaluate the pretrained model. Its omission reflects the absence of verifiable evaluation artifacts, rather than evidence about its forecasting accuracy.

Order-level market models. MarS (Li et al., 2025) and MarketGPT (Wheeler & Varner, 2024) generate order-level events within financial market simulators. MarS supports forecasting through simulated market trajectories, while MarketGPT generates order messages to model limit-order-book dynamics. Their inference workflows use order-event histories and associated market-state information that our benchmark does not supply. Our inputs instead comprise sampled scalar series, including OHLCV fields, exchange rates, macroeconomic indicators, and synthetic financial quantities. Applying these models would require additional inputs or assumptions to construct the missing order-level information, followed by aggregation of simulated events into the benchmark targets. We therefore exclude them because their input requirements do not match the common evaluation setting.

D.6 BASELINE IMPLEMENTATIONS

Foundation models. The frozen baselines use pinned public checkpoints, the shared domain contexts in Section 4, native median forecasts, and inference seed 1, without downstream fitting. Inputs are cropped to the selected context before each model's native preprocessing. Chronos-2 processes each panel jointly with cross-origin learning disabled. TimesFM 3 uses contiguous groups of at most 32 channels, retaining all targets without interaction across groups. FinCast uses the public `v1.pth` checkpoint, forecasts each channel independently, and retains released stochastic expert routing with batches of four origin–channel pairs. Our Moirai2 benchmark uses `Salesforce/moirai-2.0-R-small` through the pinned tensor forecasting interface, passing each source panel jointly. This implementation permits causal attention across channels within the panel. We use causal-mean imputation for univariate inputs and masked zero filling for multivariate inputs. Toto2 uses `Datadog/Toto-2.0-313m`, joint panel processing, native missing-value masks, and zero filling. The five baseline models retain their original decoding and other inference settings.

AR(1). The experiment selects among three configurations by domain-average validation MASE, with sMAPE breaking ties. The selected model uses ordinary least squares with an intercept and at most 512 recent finite adjacent pairs, and is refit using training and validation data. It saves raw FP64 forecasts without clipping or accuracy-based gates. Insufficient fitting data trigger persistence for 12 origin–series units; no numerical-failure substitution was recorded. Autoregressive coefficients are unconstrained.

Supervised neural models. The experiment uses TSMixer, PatchTST, and iTransformer networks trained for 3,000 updates with context 128 and learning rate 3×10^{-4}. TSMixer uses two blocks and feed-forward width 128. PatchTST and iTransformer use two encoder layers, hidden width 64, four heads, and feed-forward width 128; PatchTST uses patch length and stride 16. PatchTST is channel independent, while TSMixer and iTransformer use groups of eight channels, padding the last group and retaining all real channels. All three predict 32 steps directly and mask leads beyond the native horizon.

Training minimizes weighted per-series mean absolute error of scaled residuals, without the optional auxiliary sMAPE term. AdamW uses weight decay 10^{-4}, 200 warmup updates, gradient clipping at 1, and 64 logical origins per optimizer update. Raw forecasts add the learned residual to the last observed value, scaled by the mean absolute adjacent change over pre-origin history. The last-trained checkpoint is used for all domains, without further training, output clipping, performance gates, or domain-dependent model routing. The parameterization can yield persistence when its causal amplitude is zero or context observations are absent. We average the scores from seeds 1, 2, and 3; forecasts are not ensembled.

D.7 LoRA TRAINING AND EVALUATION

All models freeze their original weights and biases and train with context 128, seed 2027, attention rank/α 2/4, and head rank/α 8/16. All LoRA (Hu et al., 2022) runs use the same underlying benchmark training split and complete one pass through their prepared training examples. Table 2 reports results from the final checkpoints. All LoRA runs use AdamW with betas $(0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0.01, and the 5/80/15% warmup/constant/geometric-decay schedule. Peak and final learning rates are 5×10^{-6} and 5×10^{-7}. Batch sizes are selected separately for each model to accommodate differences in model size and training-memory requirements and to improve its forecast performance. The comparison therefore fixes the source-data budget and number of passes, while allowing different numbers of examples per optimizer update. Model-specific batch capacities, panel packing, and target preparation yield 8,697 updates for FTFM, 40,107 for Chronos-2, 26,843 for TimesFM 3, 108,701 for FinCast, 4,119 for Moirai2, and 20,072 for Toto2. A larger update count does not indicate additional passes over the training split or access to a larger source dataset.

The baseline quantile loss is scaled by each series' observed-context population standard deviation, using one when the scale is absent or at most 10^{-6}. Evaluation uses final adapters, the context map in Section 4.1, and inference seed 1. The resulting comparison shares training-split assignments, adapter ranks, learning rates, and per-domain evaluation contexts, while retaining the stated model-specific objectives and batching. Matching the underlying data and LoRA ranks does not imply identical target weighting, trainable adapter parameter counts, or convergence.