TIMER-M1: A MULTIVARIATE TIME SERIES FOUNDATION MODEL VIA LEARNING PRIMITIVES

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce Timer-M1, a pretrained multivariate time series foundation model that learns with primitives for zero-shot forecasting. Across domains, time series share elementary temporal and relational patterns, termed *primitives*, yet differ in how these primitives manifest and evolve across different contexts. Despite progress in zero-shot and task-general forecasting, existing foundation models may still struggle to generalize to complex real-world scenarios. To this end, we develop a primitive-based data synthesis and pretraining pipeline. The synthesis pipeline generates series with temporal primitives shared across domains and then assembles real and generated series into multivariate samples using relational primitives. Afterwards, samples are organized into *episodes* by assigning distinct channel roles as target variates, past-only covariates, and known-future covariates, ensuring that the model is optimized on predictable variates using available exogenous information. Technically, Timer-M1 further adapts gated two-dimensional Transformer blocks that dynamically allocate cross-variate attention across layers. Across three large-scale forecasting benchmarks, Timer-M1 ranks first on both FEV and TIME and second on GIFT-Eval among most recent time series foundation models. These results support effective primitive-based pretraining as a route to robust general forecasting technique across domains and task settings.

1 INTRODUCTION

Time series forecasting spans domains and task configurations in real-world applications. Time series foundation models have advanced a shift from task-specific training toward cross-domain pretraining and zero-shot forecasting (Das et al., 2024; Ansari et al., 2024; Woo et al., 2024; Khwaja et al., 2026; Liu et al., 2024b), demonstrating competitive accuracy against statistical and supervised baselines on multi-domain and multi-task benchmarks (Aksu et al., 2024; Shchur et al., 2025; Qiao et al., 2026). Meanwhile, multivariate foundation models (Ansari et al., 2025; Khwaja et al., 2026; Liu et al., 2026a) have empowered native capability to jointly predict multiple targets with historical or known-future covariates under observation settings (Shchur et al., 2025; Cohen et al., 2026).

Time series foundation models are targeted to generalize on cross-domain data and deal with heterogeneous input context. Previous foundation models have continuously improved toward this goal by task-agnostic training on real-world and synthetic datasets (Das et al., 2023; Dooley et al., 2023). However, real-world time series record complex and noisy dynamics but offer limited control over which patterns are observed; synthetic datasets provide such control but may omit the irregularities of real systems. This complementarity motivates a data construction process that combines observed dynamics with controllable relations, rather than relying on either source alone.

Recurring temporal and relational patterns provide a common basis for learning across diverse domains and forecasting tasks. Classical time series analysis separates trend, seasonal, and remainder components to understand and forecast a series (Hyndman & Athanasopoulos, 2018). We refer to such elementary patterns shared across series and domains as *primitives*: temporal primitives describe trends, periodicity, stochastic dependence, and regime changes, while relational primitives describe cross-variate correlation, delayed responses, cointegration, and event effects. These primitives characterize individual regularities rather than specific forecasting scenarios. Just as learned concepts and motifs can be generalized in new combinations in language and vision (Zhao et al., 2024;

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

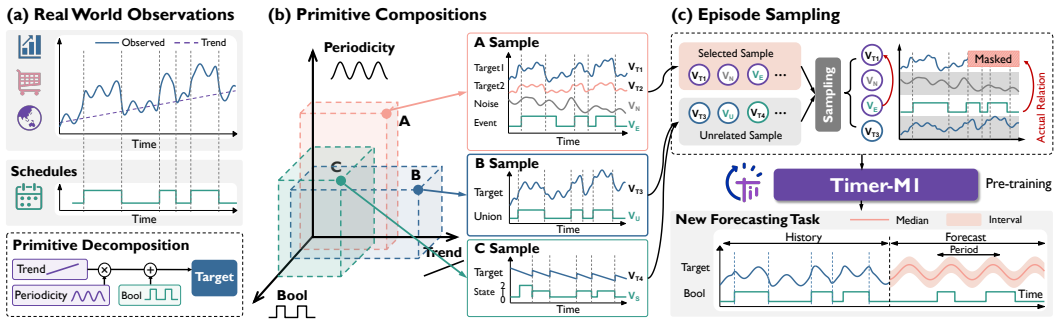

Figure 1: Learning with primitives. Temporal and relational patterns observed in real series are composed into diverse samples; episodes vary what is observed and predicted while introducing noise and interference to improve robustness, linking to generalization to new forecasting scenarios.

Liu et al., 2022), recurring primitives motivate learning transferable forecasting capabilities that remain useful when their surrounding signals and available observations change across contexts.

Learning with primitives offers a systematic route for training generalizable time series foundation models. As shown in Figure 1, we develop a compact set of operators to describe temporal patterns and cross-variate relations, and to combine synthetic and real-world time series. In addition to constructing a bottom-up data pipeline, we extend the pretraining process of foundation models into a forecasting task pipeline. We build forecasting episodes that complement metadata of real-world tasks by specifying which variates are targets and which observations are available, coupling relational structure with the information needed for forecasting. Repeated exposure under different roles and masks encourages general forecasting capabilities rather than reliance on a task configuration.

Technically, we build Timer-M1 with native zero-shot and multivariate capabilities via a primitive-based data pipeline and episode-based pretraining. Synthesis operators provide temporal and relational primitives, compose real and generated series into multivariate samples, and apply data augmentations. Episodes assign target, past-only, and known-future roles, via task-dependent supervision masks to guarantee predictability. In terms of the architecture, we adapt gated two-dimensional Transformer blocks to learning from episodes along temporal and variate dimensions, using adaptive gates learned to allocate cross-variate attention across layers. Timer-M1 ranks first on FEV and TIME and second on GIFT-Eval among the state-of-the-art foundation models; controlled primitive analyses further examine the forecasting behaviors underlying these results. Our contributions are:

- We develop a unified data synthesis pipeline that generates elementary temporal and relational primitives to enrich pretraining data and improve cross-domain generalization.
- We present Timer-M1, a time series foundation model pretrained on carefully constructed forecasting episodes that unify univariate, covariate-aware, and multivariate forecasting.
- Timer-M1 achieves strong zero-shot results across tasks, ranking first on FEV and TIME and second on GIFT-Eval, highlighting its potential for broader real-world applications.

2 Related Work

2.1 Time Series Pretraining Data

Pretraining data shape the temporal patterns and cross-variate relations a model encounters. ForecastPFN (Dooley et al., 2023) learns from synthetic tasks, Chronos (Ansari et al., 2024) combines real series with Gaussian-process compositions, and Chronos-2 (Ansari et al., 2025) constructs multivariate dependencies synthetically. Real-data curation remains complementary: TimeBench (Liu et al., 2025) supplies heterogeneous dynamics, while Timer-S1 (Liu et al., 2026b) combines corpus scaling with transformations to reduce predictive bias. Controlled studies further highlight the importance of data composition and training protocol (Wen et al., 2026), motivating coordinated data construction and task design for generalizable forecasting.

Learning from generated tasks offers a complementary approach to pretraining data construction. In the tabular domain, TabPFN (Hollmann et al., 2023; 2025) uses structural-causal-model priors, TabICL (Qu et al., 2025) scales such pretraining to larger contexts, and TabFM (Kong & Das, 2026) combines hybrid attention with causal-model generation. TabDPT (Ma et al., 2025) instead exploits signals in real tables that synthetic priors may miss. These approaches motivate combining controlled generation with empirical diversity, which we pursue by applying shared relation operators to both real and generated series, composing primitives while retaining observed dynamics.

2.2 Time Series Foundation Models

Time series foundation models explore different pretraining objectives and representations for forecasting across domains. Chronos (Ansari et al., 2024) models discretized values, whereas Sundial (Liu et al., 2025) develops continuous probabilistic generation. Autoregressive approaches learn transferable temporal patterns through next-token prediction (Liu et al., 2024b; 2026b), while TiRex (Auer et al., 2025) adopts contiguous patch masking. Beyond individual series, Moirai (Woo et al., 2024) supports variable-dimensional inputs through any-variate attention, and Timer-XL (Liu et al., 2024a) extends attention to autoregressive forecasting under multivariate and covariate settings.

Recent advances strengthen cross-variate modeling in time series foundation models. Chronos-2 (Ansari et al., 2025) combines group attention with synthetic multivariate tasks, while TiRex-2 (Podest et al., 2026) extends masked pretraining to multivariate forecasting. Toto 2.0 (Khwaja et al., 2026) combines temporal–variate attention with large-scale pretraining; TimesFM-3 (Google Research, 2026) adopts alternating attention and parallel forecasting, and Falcon-X (Liu et al., 2026a) mediates variate interaction through latent prototypes. As these architectures accommodate diverse inputs, pretraining must expose the dependencies needed to use them. This motivates us to organize primitive compositions into forecasting episodes with different variate roles and observation masks, aligning constructed relations with observation availability across task settings.

3 Approach

Timer-M1 aligns primitive-based data construction, model architecture, and pretraining tasks around learning temporal and relational primitives. Specifically, it composes primitives into multivariate data, organizes forecasting episodes, and learns from their permitted observations with gated two-dimensional Transformer blocks. We denote history length by L, forecast horizon by H, and the total number of target and covariate variates by C. The model representation width is D.

3.1 Data

Our primitive-based data pipeline brings real and synthetic series into a shared data construction process. Synthesis operators provide controlled periodic, trend, autoregressive, and other stochastic patterns. Real sources, including TimeBench (Liu et al., 2025; 2026b), retain irregular dynamics, noise, and incomplete observations that are difficult to specify exhaustively.

Relation operators accept either source origin or their combination, constructing signed responses, delayed effects, shared drivers, and other cross-variate dependencies. Thus, real observations serve not only as standalone training examples but also as inputs from which additional cross-variate relations can be generated. Transform operators derive varied responses from source series through sign inversion, resampling, clipping, local perturbations, and discrete-valued mappings. Applied during multivariate construction and subsequent augmentation, they diversify both cross-variate dependencies and the observed forms of primitives, with temporal transformations coordinated wherever alignment is required. Together, synthesis, relations, and transforms turn jointly sampled real and synthetic sources into diverse multivariate samples within a unified data construction pipeline. Original multivariate observations can also bypass synthesis to retain their existing dependencies.

Formally, let $\boldsymbol{\xi} \in \mathbb{R}^{\kappa \times \tau}$ contain κ source series aligned over τ consecutive time steps. Each row $\boldsymbol{\xi}_i$ is obtained from a real-data window or a synthesis operator $\mathcal{S}_{\gamma_i}(\boldsymbol{\varepsilon}_i)$, where γ_i specifies the generator parameters and $\boldsymbol{\varepsilon}_i$ its random inputs. A relation operator $\mathcal{R}_\phi$ produces ν dependent response series

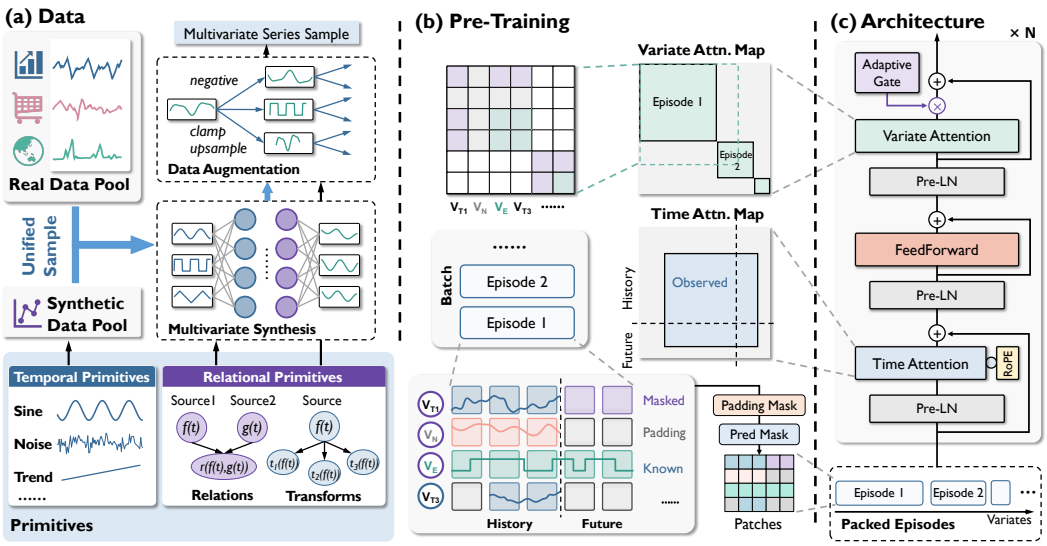

Figure 2: Timer-M1 overview. (a) Temporal and relational primitives combine real and synthetic sources into multivariate samples through synthesis and augmentation. (b) Forecasting episodes specify variate roles and observation availability, with masks controlling forecasting and within-episode attention. (c) Gated two-dimensional Transformer blocks integrate temporal and cross-variate information, with learned layer-wise gates regulating cross-variate updates.

$\boldsymbol{\eta} \in \mathbb{R}^{\nu \times \tau}$. For branches that retain the original source channels, the resulting construction is

$$\begin{aligned} \boldsymbol{\eta} &= \mathcal{R}_{\phi}(\boldsymbol{\xi}), \\ \boldsymbol{\Xi} &= \mathcal{A}_{\psi}([\boldsymbol{\xi}; \boldsymbol{\eta}])\,. \end{aligned} \tag{1}$$

Here, ϕ specifies the relation structure and its parameters, including mixing coefficients or temporal lags; $\mathcal{A}_{\psi}$ denotes the selected transforms with parameters ψ; and the semicolon denotes channel-wise concatenation. The output $\boldsymbol{\Xi}$ is an aligned multivariate record from which forecasting episodes are subsequently sampled. Response-only branches omit the original source channels from the concatenation. Generation-level dimensions κ, ν, τ are distinct from the episode dimensions C, L, H, which are determined after task-specific channel selection and temporal sampling.

For example, linear relations mix source channels with signed coefficients, whereas nonlinear relations apply a sampled mapping to the source vector. Reusing a source across multiple responses creates shared drivers, and lagged relations introduce delayed dependencies. The same construction applies whether the source dynamics are observed or synthesized. Representative constructions and the generation procedure are described in Appendix A.2.

The stable pretraining configuration references 227 dataset entries containing 214.3 million persisted records and 3.02 trillion scalar positions (Figure 3). These unweighted storage counts precede holdouts and sampling; they are not unique observations or training exposure.

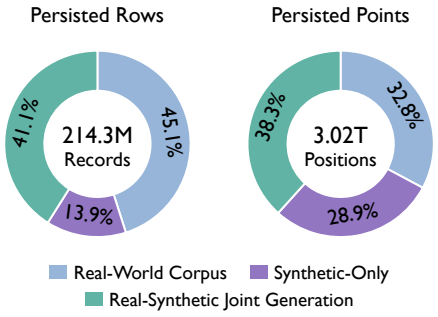

Figure 3: Persisted records and scalar positions by generation route, before sampling. Real–synthetic joint generation includes corpus-derived and mixed-source generation routes.

3.2 Architecture

Timer-M1 uses gated two-dimensional Transformer blocks to model temporal and relational primitives. Following normalization with valid-history statistics, each variate is divided into length-P patches. A residual patch embedder maps the concatenated normalized values $\widetilde{\mathbf{x}}_{c,n} \in \mathbb{R}^{P}$ and

observation mask $\mathbf{m}_{c,n} \in \{0,1\}^P$ to a D-dimensional token:

$$\mathbf{h}^0_{c,n} = \mathrm{PatchEmbed}([\widetilde{\mathbf{x}}_{c,n}; \mathbf{m}_{c,n}]), \qquad \mathbf{H}^0 \in \mathbb{R}^{C\times N\times D}, \tag{2}$$

where c indexes variates, n indexes patches, and N counts padded history and forecast patches; hidden target values and their observation masks are zeroed before embedding. Forecast tokens remain active despite their hidden values, while attention prevents interactions across episodes.

Each layer applies temporal attention, a feed-forward network, and gated variate attention in that order. With separate pre-layer normalizations LN, the residual updates are

$$\begin{aligned}
\mathbf{U}^\ell &= \mathbf{H}^\ell + \mathrm{TimeAttention}_\ell\big(\mathrm{LN}_{t,\ell}(\mathbf{H}^\ell)\big), \\
\mathbf{Z}^\ell &= \mathbf{U}^\ell + \mathrm{FFN}_\ell\big(\mathrm{LN}_{f,\ell}(\mathbf{U}^\ell)\big), \\
\mathbf{H}^{\ell+1} &= \mathbf{Z}^\ell + \alpha_\ell\, \mathrm{VariateAttention}_\ell\big(\mathrm{LN}_{v,\ell}(\mathbf{Z}^\ell)\big), \qquad \alpha_\ell = \sigma(a_\ell).
\end{aligned} \tag{3}$$

Here, $\alpha_\ell = \sigma(a_\ell)$ applies sigmoid to a learned layer-wise scalar. TimeAttention attends across patches within variates; VariateAttention attends across variates per patch. Both use multi-head scaled dot-product attention with RMS-normalized queries and keys; temporal attention uses rotary positional embeddings. The FFN comprises two linear projections with GELU. A shared quantile head maps final tokens to patch-level forecasts, restored to the original scale.

Gates adapt during training but remain fixed at inference, whereas attention weights depend on the input. Thus, the gate regulates each layer's cross-variate interaction strength rather than selecting input features. This retains standard temporal modeling while controlling the contribution of relational primitives across successive Transformer layers.

3.3 Pre-Training

An *episode* turns a generated or original multivariate record into a forecasting task by specifying history and horizon, target and covariate roles, observation availability, and future supervision. A mode describes one target and its permitted inputs; aligned modes form an episode with a shared forecast origin and attention group. Sampling different channels, windows, and roles from a record therefore exposes its underlying temporal and relational primitives under different task settings.

Write $\mathcal{E} = (\mathbf{x}, \mathbf{y}, \mathbf{c}; \mu)$, with target history $\mathbf{x} \in \mathbb{R}^{C_y\times L}$, target future $\mathbf{y} \in \mathbb{R}^{C_y\times H}$, and covariates $\mathbf{c} \in \mathbb{R}^{C_c\times(L+H)}$. Here, C_y and C_c are the numbers of target and covariate variates, with $C = C_y + C_c$. Metadata $\mu = (L, H, \mathbf{r}, \mathbf{M}^{\mathrm{obs}}, \mathbf{M}^{\mathrm{loss}}, g)$ specify lengths, variate roles $\mathbf{r}$, observation and supervision masks, and group identity g. They control packing and masking rather than supplying generating information. From permitted observations, the model predicts quantiles:

$$\widehat{\mathbf{y}}^{(q)} = f^{(q)}_\theta\big(\mathbf{x}, \mathbf{c}; L, H, \mathbf{r}, \mathbf{M}^{\mathrm{obs}}, g\big), \qquad q \in \mathcal{Q}. \tag{4}$$

Here θ denotes model parameters and $\mathcal{Q}$ the set of quantile levels. Target futures are hidden. Past-only covariates are masked after the forecast origin, whereas known-future covariates remain visible where permitted by the task setting. The same variate can take different roles across episodes. Generating annotations guide the selection of derived targets and source or auxiliary variates, exposing direct, shared, and partially observed relations without revealing operator identities to the model. Additional channels may be redundant or unrelated to the prediction target. Original multivariate observations use the same interface without requiring explicit generating annotations.

Temporal sampling varies both the observed span of a primitive and the extent of its extrapolation. Episodes are packed with separate attention groups and optimized in three steps. First, define a binary supervision mask M^{loss}_{ih}, equal to one only at valid target-future entries, and apply horizon-decaying weights over the padded forecast length $H_{\mathrm{pad}} \geq H$:

$$w_h = \frac{h^{-1}}{H^{-1}_{\mathrm{pad}} \sum_{j=1}^{H_{\mathrm{pad}}} j^{-1}}, \qquad \omega_{ih} = M^{\mathrm{loss}}_{ih} w_h. \tag{5}$$

Second, average the pointwise pinball loss over quantiles, using targets and predictions normalized by observed-history statistics:

$$\ell_{ih} = \frac{2}{|\mathcal{Q}|} \sum_{q\in\mathcal{Q}} \rho_q\left(\widetilde{y}_{ih} - \widetilde{\widehat{y}}^{(q)}_{ih}\right), \qquad \rho_q(v) = v\big(q - \mathbf{1}[v < 0]\big). \tag{6}$$

For each target variate, tildes denote normalization of both targets and predictions using the same statistics computed from its observed history. Finally, aggregate over target variates i and forecast positions h:

$$\mathcal{L}(\theta) = \frac{\sum_{i,h} \omega_{ih} \ell_{ih}}{\sum_{i,h} \omega_{ih} + \epsilon}. \tag{7}$$

Here $\epsilon = 10^{-8}$ stabilizes the denominator. This objective excludes unobserved supervision and emphasizes nearer forecast positions without changing the observation mask. Large-scale multivariate pretraining is followed by long-context post-training. Together, the shared objective and varied episode metadata train forecasting across different primitive compositions and observation availability. An optional GroupMix extension is described in Appendix B.3.

4 Experiments

We evaluate Timer-M1 through public benchmarks, primitive evaluation, and ablation studies. These examine forecasting accuracy, temporal extrapolation, relational forecasting, and input reliability, alongside the contributions of adaptive gates and data construction. Together, they connect performance with learning reusable primitives through coordinated data construction, episode-based pretraining, and model design.

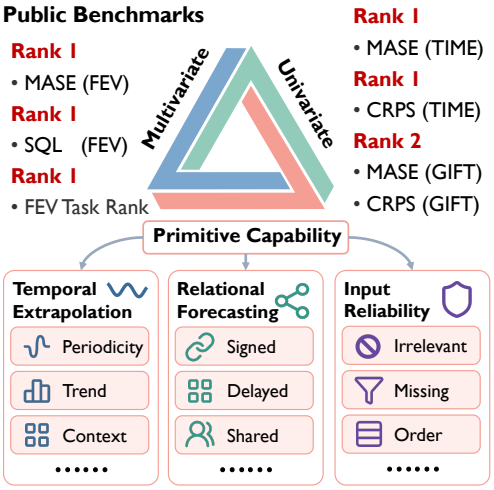

Figure 4: Evaluation overview. Public benchmarks assess forecasting across settings, while primitive evaluations examine temporal extrapolation, relational forecasting, and input reliability. Rankings are among compared methods.

4.1 Public Benchmarks

We evaluate 100 FEV (Shchur et al., 2025), 97 GIFT-Eval (Aksu et al., 2024), and 98 TIME (Qiao et al., 2026) configurations, reporting point and probabilistic scores over matched sets. Public baseline submissions are compared with a shared Timer-M1 checkpoint. FEV distinguishes five task types, while TIME and predominantly univariate GIFT-Eval complement coverage across temporal settings. Details are provided in Appendix B.4.

FEV. Timer-M1 achieves MASE skill 0.3771 and SQL skill 0.4887 over the 100 matched configurations, ranking first on both aggregate metrics among the compared methods. Its relative errors are lower than those of Chronos-2 by 3.43% and 3.02%, and lower than those of TimesFM-3 by 0.47% and 0.42%, respectively. Figure 5 further separates five mutually exclusive task types: univariate, multivariate, past-only covariates, known-future covariates, and their combination. Across these groups, Timer-M1 achieves the best unweighted average rank for both MASE (1.80) and SQL (1.60), placing among the top two models in every group under both metrics. This consistency complements its aggregate accuracy, demonstrating broad task coverage within a compact 121M-parameter backbone. A shared frozen checkpoint and inference recipe accommodate different variate roles and observation conditions without task-specific parameter updates, connecting temporal and relational primitives with flexible episode construction.

TIME. Timer-M1 obtains normalized MASE 0.6372 and CRPS 0.5363 over 98 configurations (Figure 6), reducing errors relative to Chronos-2 by 3.75% and 3.60%, respectively. These improvements extend across both point and probabilistic aggregate scores, suggesting benefits beyond point estimation alone, complementing the results on FEV across diverse forecasting task settings. For comparison with TimesFM-3, a repeat under the same evaluation conditions yields 0.63977 and 0.53635, corresponding to reductions of 0.40% and less than 0.01%. We interpret this narrow margin as competitive performance with TimesFM-3 rather than statistically established superiority, while observing larger improvements over Chronos-2 on both reported metrics.

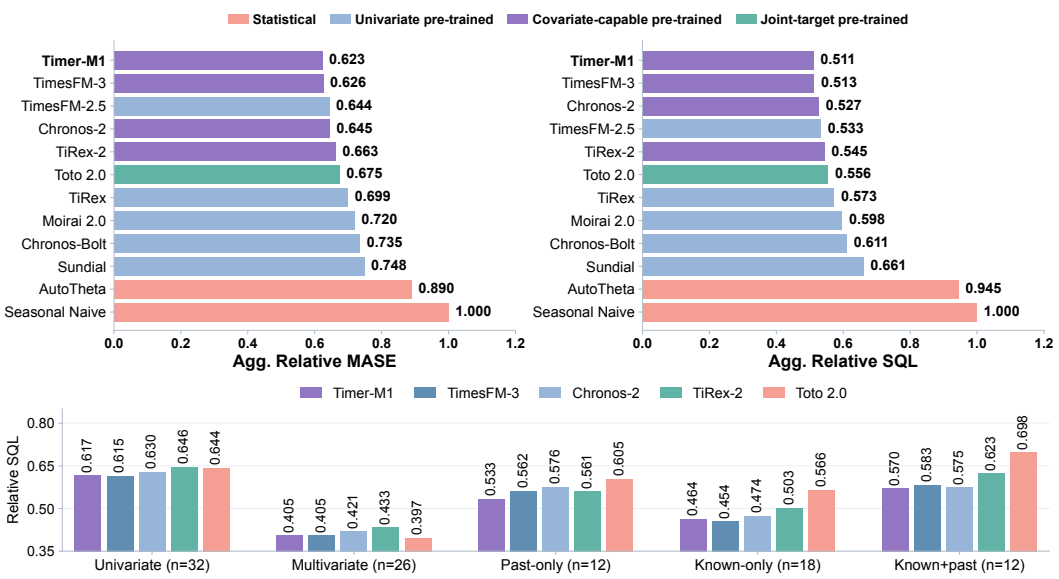

Figure 5: Forecasting results on FEV across 100 matched configurations. Top: aggregate MASE and SQL for 12 methods, normalized by Seasonal Naive. Bottom: SQL across five task types with different variate roles and observation availability. Timer-M1 achieves the lowest aggregate errors. Colors denote forecasting interfaces above and individual models below.

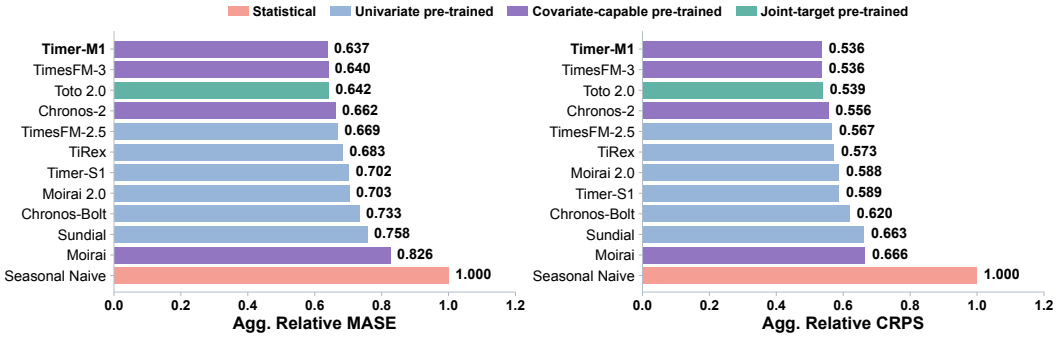

Figure 6: Comparison on TIME across 98 matched configurations. Timer-M1 achieves state-of-the-art aggregate performance, attaining the lowest normalized MASE and CRPS among the compared methods across diverse forecasting task settings. Colors distinguish forecasting interfaces.

GIFT-Eval. Across 97 configurations, Timer-M1 obtains normalized MASE 0.6806 and CRPS 0.4688 (Figure 7), reducing errors relative to Chronos-2 by 2.47% and 3.43%, respectively. Timer-M1 also improves both metrics over TiRex-2 and Toto 2.0 (2.5B), ranking second among the compared methods, while TimesFM-3 retains the lowest aggregate errors. The benchmark spans diverse sampling frequencies and forecast lengths, testing whether a shared pretrained model remains effective across heterogeneous temporal resolutions and task settings. These results on predominantly univariate GIFT-Eval complement FEV and TIME, showing that Timer-M1's improvements extend beyond multivariate tasks to both point and probabilistic forecasting of individual series.

4.2 Primitive Evaluation

Primitives encompass recurring temporal and relational patterns that support forecasting across task settings. To examine how Timer-M1 learns and applies these primitives, we conduct *Primitive Evaluation*, assessing temporal extrapolation, relational forecasting, and input reliability through controlled tasks. Figure 8 presents aggregate results over complete task groups alongside illustrative cases, connecting forecasting performance with responses to systematic changes in temporal patterns, variate relations, and observation availability, thereby examining primitive reuse under changing conditions. Details are provided in Appendices B.5, C.3, and D.

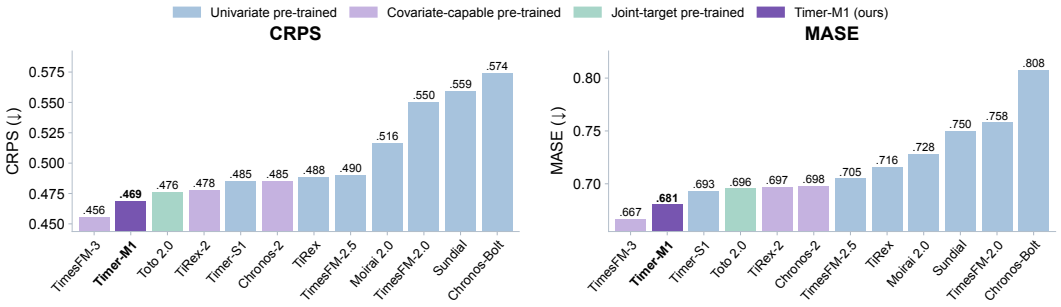

Figure 7: Point and probabilistic forecasting performance on GIFT-Eval across 97 matched configurations. Timer-M1 ranks second among 12 compared pretrained models on both MASE and CRPS, demonstrating competitive generalization across diverse domains and forecast horizons.

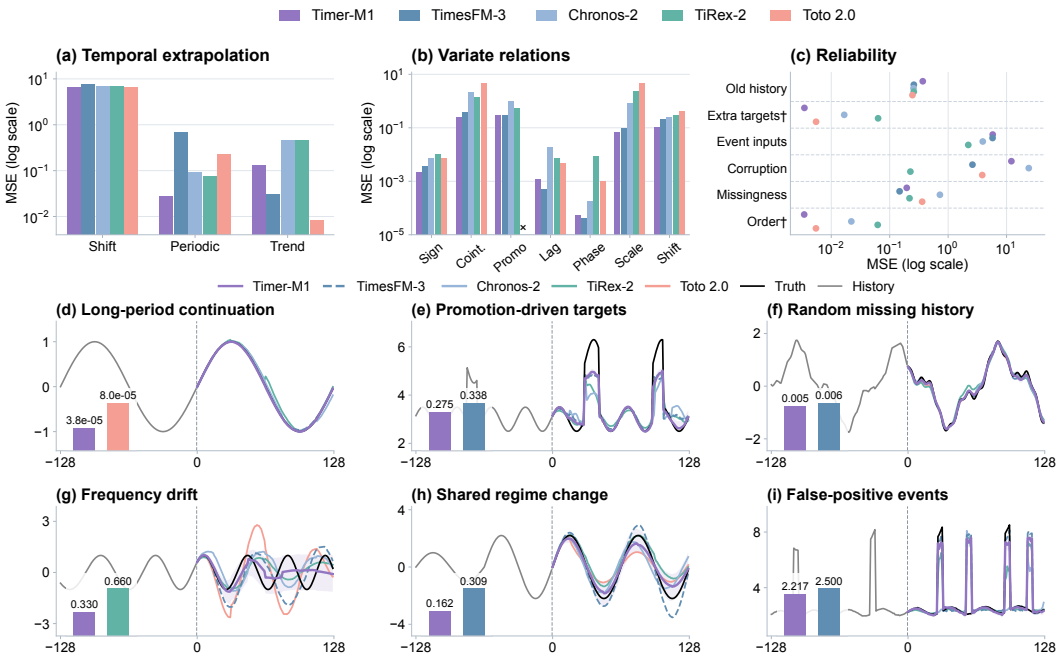

Figure 8: Primitive evaluation. Top: predefined temporal groups, seven relational mechanisms, and six reliability categories, measured by MSE (lower is better). Unsupported interfaces are omitted; detailed results for the displayed groups appear in Appendix C.3. Bottom: six strength cases spanning distinct mechanisms, not an estimate of their frequency. Shading denotes M1's marginal 10–90% quantiles. † marks unavailable squared-error archives for some baseline.

Temporal extrapolation. Timer-M1 shows strong periodic extrapolation, achieving NMAE 0.02432 across 13 periodic cases. Long-period continuation and frequency drift illustrate its ability to follow recurring patterns beyond the observed context. These results are consistent with learning reusable temporal primitives across diverse compositions, linking accurate periodic continuation with the ability to track evolving temporal patterns rather than repeat observed sequences.

Relational forecasting. Timer-M1 achieves NMAE 0.02160 across six joint-target tasks, demonstrating strong cross-variate forecasting. Promotion-driven targets and shared regime changes illustrate how relational primitives connect observations to target dynamics. These findings support learning reusable relational primitives through data construction and episode-based pretraining, which exposes dependencies under different variate roles and observation availability.

Input reliability. Timer-M1 reduces target-missingness NMAE by 32.5% relative to Chronos-2, with selected cases illustrating resilience to incomplete histories and misleading inputs. These

findings support composing primitives under varied observation conditions to broaden coverage of challenging forecasting scenarios. Episodes with observation masks and redundant or unrelated variates complement this construction by exposing primitives through imperfect inputs.

4.3 Ablation Study

We conduct ablation studies to examine how adaptive cross-variate interaction and primitive-based data construction contribute to forecasting. Gate ablations compare learned gates with fixed alternatives, testing the benefit of allocating interaction strength across layers. Data ablations compare four recipes to assess the contribution of joint real–synthetic generation beyond source diversity. All variants use the same evaluation inputs and inference recipe. Figure 9 reports overall and task-specific FEV results, while Table 1 summarizes data ablations across benchmarks.

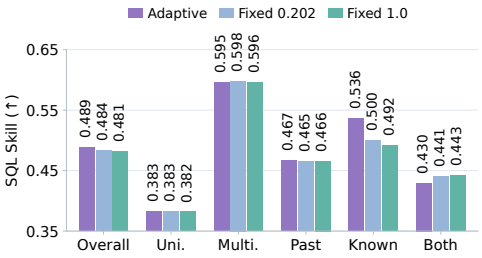

Figure 9: Gate ablation on overall FEV and five task types. Improvement supports layer-wise allocation of cross-variate interaction.

Metric	Real Only	Synthetic Only	No-Joint	Full Recipe
FEV Skill Score ↑	0.4431	0.4710	0.4725	**0.4887**
GIFT MASE ↓	0.6998	0.6884	0.6819	**0.6806**
GIFT CRPS ↓	0.4850	0.4741	0.4722	**0.4688**
TIME MASE ↓	0.6579	0.6438	0.6379	**0.6372**
TIME CRPS ↓	0.5526	0.5456	**0.5359**	0.5363

Table 1: Data recipe ablation. Red bold and blue underline mark the best and second-best scores in each row. Joint generation benefits most aggregates.

Adaptive gates. We compare learned layer-wise coefficients α_ℓ with two uniform alternatives. *Fixed 0.202* uses the mean gate value from the stable checkpoint, preserving a representative interaction strength while removing layer-wise variation; *Fixed 1.0* applies unattenuated cross-variate updates. Adaptive gates achieve FEV SQL skill 0.4887, versus 0.4836 and 0.4814, improving three of five task types. The learned α_ℓ generally increase with depth, suggesting stronger cross-variate integration in deeper layers. Together, this depth-dependent pattern and the advantage over mean-matched scaling support allocating interaction strength across layers, allowing relational primitives to be integrated progressively with temporal representations.

Data construction. We compare *Real Only*, *Synthetic Only*, *No-Joint*, and the *Full Recipe*. The first two restrict generation to their respective source pools. No-Joint includes both sources but excludes joint real–synthetic generation; the full recipe additionally composes them through shared relation operators. The full recipe improves FEV skill and both GIFT-Eval metrics over all three alternatives and achieves the lowest TIME MASE. Its FEV skill exceeds No-Joint by 0.0162, suggesting benefits beyond source diversity alone. By combining observed dynamics with controlled relational primitives, joint generation broadens multivariate compositions, supporting a unified data construction pipeline rather than independent real and synthetic training collections.

5 Conclusion

We present Timer-M1, a multivariate time series foundation model that learns with temporal and relational primitives. Its primitive-based data pipeline combines observed dynamics with controlled generation to construct diverse multivariate samples. Episode-based pretraining exposes these primitives under varied variate roles and observation availability, while gated two-dimensional Transformer blocks integrate temporal and cross-variate information. Evaluations on three public benchmarks and controlled primitive tasks demonstrate competitive forecasting performance and strengths in temporal extrapolation and relational forecasting. These results support aligning data construction, forecasting episodes, and model architecture around shared primitives. Future work will expand real-data coverage and primitive compositions to better represent irregular dynamics and partially observed dependencies in complex real-world systems. On the architectural side, exploring input-dependent cross-variate interaction and more efficient long-context modeling may further improve how foundation models learn and apply primitives across domains and task settings.

AI USE STATEMENT

Generative AI tools were used only for translation, drafting and revising portions of the manuscript, language polishing, literature search, reference formatting, and figure preparation. They were not used for developing theoretical models or conceptual frameworks, designing research methodology or experiments, writing proofs, implementing experimental methods, generating or processing datasets, conducting data analysis, or interpreting results, and no substantive AI-generated suggestions regarding mathematical assumptions, derivations, proofs, or research hypotheses were adopted. All AI-assisted work was reviewed by the authors, who consulted original references, checked derivations and experimental code, and verified reported results against experimental records; we take full responsibility for the final content of this work.

REFERENCES

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. GIFT-Eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024.

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

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, et al. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

Andreas Auer, Patrick Podest, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. Tirex: Zero-shot forecasting across long and short horizons with enhanced in-context learning. *arXiv preprint arXiv:2505.23719*, 2025.

Ben Cohen, Emaad Khwaja, Youssef Doubli, Salahidine Lemaachi, Chris Lettieri, Charles Masson, Hugo Miccinilli, Elise Ramé, Qiqi Ren, Afshin Rostamizadeh, et al. This time is different: An observability perspective on time series foundation models. *Advances in neural information processing systems*, 38:50907–50951, 2026.

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

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

Samuel Dooley, Gurnoor Singh Khurana, Chirag Mohapatra, Siddartha Naidu, and Colin White. Forecastpfn: Synthetically-trained zero-shot forecasting. *arXiv preprint arXiv:2311.01933*, 2023.

Google Research. TimesFM 3.0: Official model and evaluation repository. `https://github.com/google-research/timesfm`, 2026. Accessed September 12, 2026.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations*, 2023. URL `https://arxiv.org/abs/2207.01848`.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637:319–326, 2025. doi: 10.1038/s41586-024-08328-6. URL `https://doi.org/10.1038/s41586-024-08328-6`.

Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

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`.

Weihao Kong and Abhimanyu Das. Introducing TabFM: A zero-shot foundation model for tabular data. Google Research Blog, 2026. URL https://research.google/blog/introducing-tabfm-a-zero-shot-foundation-model-for-tabular-data/. Published June 30, 2026. Accessed September 16, 2026.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, 2022. URL https://arxiv.org/abs/2206.01714.

Yiding Liu, Yifan Hu, Hongjie Xia, Peiyuan Liu, Hongzhou Chen, Xilin Dai, Zewei Dong, and Jiang-Ming Yang. Falcon-X: A time series foundation model for heterogeneous multivariate modeling. *arXiv preprint arXiv:2605.27286*, 2026a. URL https://arxiv.org/abs/2605.27286.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-xl: Long-context transformers for unified time series forecasting. *arXiv preprint arXiv:2410.04803*, 2024a.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *Forty-first International Conference on Machine Learning*, 2024b.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. *arXiv preprint arXiv:2502.00816*, 2025.

Yong Liu, Xingjian Su, Shiyu Wang, Haoran Zhang, Haixuan Liu, Yuxuan Wang, Zhou Ye, Yang Xiang, Jianmin Wang, and Mingsheng Long. Timer-S1: A billion-scale time series foundation model with serial scaling. *arXiv preprint arXiv:2603.04791*, 2026b. URL https://arxiv.org/abs/2603.04791.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Hamidreza Kamkari, Jesse C. Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L. Caterini, and Maksims Volkovs. TabDPT: Scaling tabular foundation models on real data. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2410.18164.

Patrick Podest, Marco Pichler, Elias Bürger, Levente Zólyomi, Bernhard Voggenberger, Wilhelm Berghammer, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. TiRex-2: Generalizing TiRex to multivariate data and streaming. *arXiv preprint arXiv:2607.01204*, 2026. URL https://arxiv.org/abs/2607.01204.

Zhongzheng Qiao, Sheng Pan, Anni Wang, Viktoriya Zhukova, Yong Liu, Xudong Jiang, Qingsong Wen, Mingsheng Long, Ming Jin, and Chenghao Liu. It's TIME: Towards the next generation of time series forecasting benchmarks. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.12147.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *Proceedings of the International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2502.05564.

Oleksandr Shchur, Abdul Fatir Ansari, Caner Turkmen, Lorenzo Stella, Nick Erickson, Pablo Guerron, Michael Bohlke-Schneider, and Yuyang Wang. fev-bench: A realistic benchmark for time series forecasting. *arXiv preprint arXiv:2509.26468*, 2025.

Yunshi Wen, Wesley M. Gifford, Chandra Reddy, Lam M. Nguyen, Jayant Kalagnanam, and Anak Agung Julius. Revisiting the generic transformer: Deconstructing a strong baseline for time series foundation models. *arXiv preprint arXiv:2602.06909*, 2026. URL https://arxiv.org/abs/2602.06909.

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

Haoyu Zhao, Simran Kaur, Dingli Yu, Anirudh Goyal, and Sanjeev Arora. Can models learn skill composition from examples? In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2409.19808.

A Dataset Statistics

This section describes the temporal and relational primitives used by the data construction pipeline. The discussion complements Section 3.1 by connecting operator families to the temporal evolution and cross-variate dependencies that define a forecasting sample. Real observations retain empirical dynamics, while generated series make selected patterns and dependencies controllable. Both enter the same multivariate construction process, allowing empirical dynamics and controlled primitive compositions to contribute to a shared source of forecasting episodes.

A.1 Temporal and Relational Primitives

Temporal primitives describe evolution within a series. Periodic and trend components may appear alone or together with stochastic variation; autoregressive dependence provides a further source of temporal continuity. Relational primitives describe how multiple variates evolve together. Source reuse introduces shared drivers, signed mixing produces positive or negative responses, and temporal offsets create delayed effects. These families are complementary: temporal primitives determine the evolution of individual sources, while relational primitives determine how their information is shared across variates. For example, a periodic source can produce responses with different signs, scales, or delays without changing its underlying temporal pattern. The resulting dependencies are specified during construction rather than inferred as causal mechanisms from real observations.

Table 2: Operator families in primitive-based data construction. The taxonomy follows the temporal and relational primitives introduced in the main text. Examples describe mechanisms rather than a fixed generation recipe.

Family	Primitive	Representative construction
Temporal	Periodicity	Oscillatory components and their superposition
	Trend	Gradual growth, decline, and shared drift
	Stochastic dependence	Noise and autoregressive dynamics
	Regime changes	Piecewise changes in level or evolution
Relational	Signed responses	Positive and negative source mixing
	Delayed responses	Temporal offsets between source and response
	Shared drivers	Multiple responses derived from common sources
	Cointegration	Shared nonstationary components
	Event effects	Responses to discrete-valued inputs
	Transforms	Inversion, clipping, resampling, and discrete mappings

A.2 Construction and Observation

Following Equation 1, source series are selected from real observations or generated temporal primitives, combined through relation operators, and transformed into aligned multivariate records. For source array $\boldsymbol{\xi} \in \mathbb{R}^{\kappa \times \tau}$, a schematic linear response is

$$\eta_i(t) = \sum_{j \in \mathcal{J}_i} w_{ij} \xi_j(t - \delta_{ij}), \qquad i = 1, \ldots, \nu, \tag{8}$$

where $\mathcal{J}_i \subseteq \{1, \ldots, \kappa\}$ selects the sources contributing to response i, w_{ij} controls their signed contribution, and δ_{ij} denotes a temporal lag. Nonlinear relation operators replace the linear combination with a mapping of the selected sources. Reusing a source across responses introduces shared variation. The transform operator $\mathcal{A}_\psi$ then changes the observed forms through operations such as rescaling or clipping. For branches retaining source channels, the resulting record is $\boldsymbol{\Xi} = \mathcal{A}_\psi([\boldsymbol{\xi}; \boldsymbol{\eta}])$; response-only branches omit $\boldsymbol{\xi}$. Transformations that modify time coordinates are coordinated where alignment must be preserved. Original multivariate observations can also enter directly, retaining their existing dependencies.

A multivariate record specifies aligned values, whereas a forecasting episode additionally specifies their roles and availability. After channel selection and temporal sampling, let $\mathbf{X} \in \mathbb{R}^{C \times (L+H)}$

contain the episode's target and covariate values. Following the main text, $\mathbf{M}^{\text{obs}} \in \{0,1\}^{C\times(L+H)}$ denotes the observation mask, while $\mathbf{M}^{\text{loss}} \in \{0,1\}^{C_y\times H}$ selects supervised target-future entries. The masked values are defined entrywise as

$$X^{\text{obs}}_{c,t} = \begin{cases} X_{c,t}, & M^{\text{obs}}_{c,t} = 1, \\ 0, & M^{\text{obs}}_{c,t} = 0, \end{cases} \qquad M^{\text{loss}}_{ih} = \begin{cases} 1, & \text{if target } i \text{ at horizon } h \text{ is valid,} \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Here, c indexes all episode variates, t indexes history and forecast positions, i indexes target variates, and h indexes forecast positions. Normalization uses observed-history statistics, and hidden entries remain zero in the values supplied to the patch embedder. The observation mask accompanies these values so that hidden entries remain distinguishable from observed zeros. Target futures are withheld from the input but may contribute to supervision; known-future covariates remain visible where permitted. Different role assignments therefore expose the same primitives through different forecasting conditions without changing the underlying record.

The record and scalar-position counts in Figure 3 describe stored data before episode sampling. Records, channel-wise series, and scalar positions are distinct counting units. If record r contains C_r channels of length T_r, their respective counts are

$$N_{\text{records}} = \sum_r 1, \qquad N_{\text{series}} = \sum_r C_r, \qquad N_{\text{positions}} = \sum_r C_r T_r. \tag{10}$$

These quantities characterize stored data volume rather than independent forecasting tasks or consumed training tokens. Episode sampling can reuse a record under different horizons and observation roles.

B Implementation Details

B.1 Model Configuration

Timer-M1 uses the same gated two-dimensional Transformer backbone across task settings. Table 3 summarizes its dimensions. Temporal attention operates along patches within each variate, and variate attention operates across variates at each patch position. The residual order is temporal attention, feed-forward transformation, and gated variate attention. The prediction head outputs marginal quantiles, with the median used for point forecasts.

Table 3: Model configuration. The gate contributes one learned scalar per layer; the total parameter count includes the embedding and prediction head.

Component	Configuration
Parameters	120,998,412 (approximately 121M)
Transformer layers	12
Representation / feed-forward width	768 / 3072
Attention heads / head dimension	12 / 64
Patch length	16 observations
Prediction quantiles	$0.1, 0.2, \ldots, 0.9$
Temporal position encoding	Rotary positional embeddings
Gate parameterization	$\alpha_\ell = \sigma(a_\ell)$; one scalar per layer
Gate initialization	$a_\ell = -4.6$, giving $\alpha_\ell \approx 0.01$

For C variates, N patches, and width D, factorized attention requires $\mathcal{O}(CN^2D + NC^2D)$ operations per layer, rather than $\mathcal{O}(C^2N^2D)$ for attention over all flattened tokens. Linear projections and feed-forward transformations additionally require $\mathcal{O}(CND^2)$ operations. Gating adds a scalar multiplication to each variate-attention residual update. The same patch embedder and prediction head are shared across variates, while episode boundaries determine which cross-variate interactions are permitted. This separates the number and roles of input variates from the learned representation width.

B.2 Layer-wise Gates

Figure 10 shows the learned coefficients $\alpha_\ell = \sigma(a_\ell)$ of the evaluated model. Each coefficient scales the variate-attention residual update after temporal attention and the feed-forward transformation. It is shared across tokens in a layer and remains fixed during inference; it is not a sample-dependent attention score. The coefficients generally increase with depth, with larger updates in later layers and nonmonotonic local variation. This pattern is consistent with first forming temporal representations and subsequently giving greater weight to cross-variate integration, although coefficient magnitudes alone do not measure the information carried by either branch.

The fixed value 0.202 is the rounded mean of twelve gate coefficients from an earlier stable model. It preserves a representative interaction scale while removing layer-specific allocation; the other fixed control, 1.0, uses the unscaled variate-attention update. As shown in Table 11, the learned gates improve overall FEV skill.

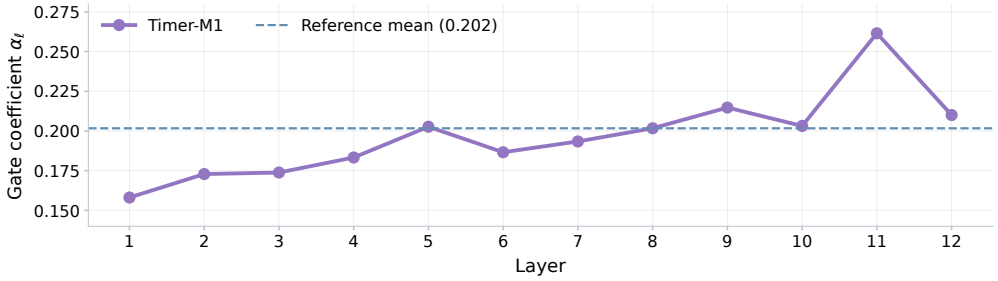

Figure 10: Layer-wise allocation of cross-variate interaction. The learned coefficients generally rise with depth, while local deviations allow different layers to assign different weights to variate-attention updates. The dashed line marks the earlier stable-model mean used by the fixed-0.202 control, not the mean of the displayed curve. Together with the gate ablation, the profile motivates adaptive allocation rather than a uniform residual scale.

B.3 Episodes and Packing

Episodes share a forecast origin and specify target, past-only covariate, and known-future covariate roles. Target futures are hidden; available future covariates remain visible. Observation masks determine which values enter the model, while supervision masks select valid target-future entries for the loss. Padding is excluded from valid observations without disabling the forecast tokens that must produce predictions.

We formulate episode batching as a capacity-constrained bin-packing problem and use a greedy packing heuristic. Episodes remain intact, and distinct attention groups prevent information exchange between packed episodes. Packing changes how episodes share a batch, not their permitted observations or prediction targets. The packing objective improves batch utilization while preserving episode semantics; the greedy procedure does not require an exact solution of the bin-packing problem.

GroupMix. GroupMix extends episode construction by changing cross-variate visibility during training. Starting from separate episode groups, a selected group is associated with additional variates from other groups in the packed batch. In the group-level form, the selected groups share an attention group, permitting interaction among their visible variates. A finer-grained form introduces only a subset of external variates. Sequence values, variate roles, and supervision targets are retained: the operation modifies the attention grouping rather than interpolating series or mixing forecast labels.

Observation and supervision masks continue to enforce the original information constraints. In particular, hidden target futures are not made visible by a group reassignment, and padded entries do not become observations. Unrelated external variates can consequently act as distractors, providing a way to vary the context in which temporal and relational primitives must be recognized. GroupMix is a training-time extension; ordinary episode grouping is used at inference. It is not enabled for the

results reported here, so its potential regularization effect is distinct from the measured gate and data-construction ablations.

B.4 Public Benchmark Protocols

We follow the official task definitions and scoring conventions of FEV, TIME, and GIFT-Eval (Shchur et al., 2025; Qiao et al., 2026; Aksu et al., 2024). Forecast origins, prediction horizons, and permitted observations are determined by the benchmark tasks. FEV distinguishes historical covariates from future-known covariates; only inputs available at the forecast origin are supplied. Scores use the complete matched configuration sets reported in the main text. Public reference scores are fixed comparison snapshots, not a continuously updated ranking.

The official online evaluation documentation was checked on September 26, 2026.[1] Compliance with task definitions and metrics is distinct from training-data eligibility. No task-specific parameter updates are performed during evaluation. Training-data exposure is not matched across the compared methods.

For model m and configuration j, let $E_{m,j}$ denote the benchmark error and $E_{\mathrm{SN},j}$ the Seasonal Naive error. For a configuration group G, the relative aggregate is

$$R_{m,G} = \exp\left(\frac{1}{|G|}\sum_{j\in G}\log\frac{E_{m,j}}{E_{\mathrm{SN},j}}\right). \tag{11}$$

FEV clips each relative error to $[0.01, 100]$ before aggregation and reports skill $1 - R_{m,G}$. We report MASE and SQL skill separately. TIME and GIFT-Eval report normalized MASE and quantile-based CRPS. Higher skill and lower normalized error indicate better forecasts. Missing matched results remain unavailable rather than being assigned zero error.

B.5 Primitive Evaluation Protocol

The controlled evaluation separates temporal extrapolation, relational forecasting, and input reliability. Each test fixes a generated trajectory and forecast origin, then specifies the observations available to each model. All compared models receive the same target values and permitted covariates for a given setting. Future targets are used only for scoring. The displayed groups are defined by their mechanism, not by the model attaining the smallest error, and retain all scoring records within each selected group.

Temporal extrapolation. The elementary-function tests use $L = 256$ historical observations and $H = 128$ forecast steps. Periodicity covers sinusoidal and nonsinusoidal waves, multiple frequencies, amplitude modulation, frequency drift, and narrow pulses. Trend tests include constant, increasing, decreasing, nonlinear, and piecewise trajectories. The shift group includes a level change visible in history and changes in level, slope, amplitude, frequency, or phase introduced at the forecast boundary. The latter have no advance indication in the supplied history and are interpreted as stress tests, not as evidence that a model can anticipate an unobserved intervention. Table 4 connects these mechanisms to the result groups.

Relational forecasting. Structured tasks use the same $L = 256$, $H = 128$ window and construct multiple responses from shared sources. Signed, delayed, phase-shifted, scale-varying, cointegrated, and shared-regime relations probe different ways of transferring information across variates. The joint-target settings hide all target futures; promotion-driven settings additionally supply the event covariate over its permitted interval. Each target is scored separately before group aggregation. These cross-model comparisons assess forecasting with the specified inputs; they are not, by themselves, estimates of the gain from joint rather than independent prediction.

Input reliability. These tests use $L = 512$ and $H = 128$ and hold the clean future fixed while changing the available history or auxiliary inputs. Missing-history settings include random removal at 10%, 30%, and 50%, internal gaps of 32 or 96 steps, and missing suffixes of 16 or 64 steps.

[1] FEV, TIME, and GIFT-Eval.

Internal gaps use common linear interpolation; missing suffixes use forward filling, without consulting future targets. Corruption tests perturb the observed history; irrelevant-input tests introduce unrelated or misleading context. Event tests alter the supplied event indicators while retaining the same target trajectory. Variate-order tests permute the same input channels and restore predictions to the original target order before scoring. Thus, missingness results concern forecasting after the specified common preprocessing, rather than a separate imputation task.

Table 4: Displayed primitive evaluation groups and their controlled conditions. The counts of scoring records and model-specific results appear in Tables 8–10. All groups report MSE; NMAE additionally supports the comparisons discussed in the text.

Capability	Group	Controlled pattern or observation condition
Temporal	Periodic	Period, waveform, modulation, and gradual frequency change
	Trend	Direction, growth shape, and observed slope change
	Shift	Observed level change and unannounced boundary changes
Relational	Sign; scale	Shared dynamics with opposite signs or different scales
	Lag; phase	Delayed responses and phase-shifted periodic targets
	Cointegration	Multiple responses sharing a nonstationary source
	Promotion; shift	Event-driven responses and shared regime changes
Reliability	Irrelevant context	Uninformative or misleading portions of historical context
	Irrelevant targets	Additional unrelated variates in the joint target set
	Missing event inputs	Incomplete or incorrect future event information
	Target corruption	Noise and outlying values in the observed target history
	Target missingness	Random gaps, internal blocks, and missing recent history
	Variate order	Permutation of input channels with target identities restored

Metrics and aggregation. For scoring record r, let Ω_r contain its evaluated forecast positions and let $\hat{y}_{r,i}$ and $y_{r,i}$ denote the point forecast and ground truth. We compute

$$\mathrm{MSE}_r = \frac{1}{|\Omega_r|} \sum_{i \in \Omega_r} (\hat{y}_{r,i} - y_{r,i})^2, \qquad \mathrm{NMAE}_r = \frac{\sum_{i \in \Omega_r} |\hat{y}_{r,i} - y_{r,i}|}{|\Omega_r| s_r}, \tag{12}$$

where s_r is a fixed reference scale shared by the models for that record. The displayed short-window temporal and relational groups use the range of the complete generated trajectory, including history and future. For a nearly constant trajectory, the denominator is the larger of its mean absolute value and one. Reliability tests use the range of the clean historical reference with a positive floor of 10^{-6}. These quantities normalize evaluation only and are not provided to the forecasting model. Group scores are arithmetic means over their scoring records, with one record per evaluated target and condition. The count n is therefore not necessarily the number of independent generated trajectories. Rankings are computed before rounding; red bold and blue underline indicate the lowest and second-lowest distinct errors, respectively.

MSE retains the effect of absolute error scale, whereas NMAE facilitates comparison between trajectories of different amplitudes. The two metrics are not interchangeable, and no cross-capability composite score is inferred from the selected groups. Unsupported input interfaces and unavailable metrics are left unranked rather than assigned an artificial penalty. Where quantiles are available, showcase shading spans the 10th to 90th percentiles; these are marginal prediction intervals, not confidence intervals for the estimated mean. Examples illustrate individual mechanisms and do not determine the numerical aggregates.

C Full Results

C.1 Public Benchmarks

Table 5 provides the numerical aggregates behind the public-benchmark comparisons. Scores retain the original benchmark direction and normalization. The comparisons cover 100 FEV, 98 TIME, and 97 GIFT-Eval configurations. Training-data eligibility is described in Appendix B.4; numerical ranks refer to the displayed methods rather than a claim of official zero-shot leaderboard placement.

Table 5: Aggregate benchmark results. FEV reports skill (↑); TIME and GIFT-Eval report normalized errors (↓). Red bold and blue underline mark the best and second-best displayed scores per column. Dashes indicate unavailable matched results. Training exposure differs across methods (Appendix B.4).

Model	FEV MASE	FEV SQL	TIME MASE	TIME CRPS	GIFT MASE	GIFT CRPS
Timer-M1	**0.3771**	**0.4887**	**0.6372**	**0.5363**	0.6806	0.4688
TimesFM-3	0.3742	0.4866	0.6398	0.5363	**0.6668**	**0.4557**
Chronos-2	0.3550	0.4728	0.6620	0.5563	0.6978	0.4854
TiRex-2	0.3374	0.4550	–	–	0.6973	0.4781
Toto 2.0 (2.5B)	0.3254	0.4442	0.6419	0.5394	0.6956	0.4759
TimesFM-2.5	0.3562	0.4668	0.6686	0.5674	0.7050	0.4903
Timer-S1	–	–	0.7021	0.5886	0.6934	0.4853
Sundial	0.2522	0.3387	0.7576	0.6634	0.7502	0.5590
Moirai 2.0	0.2803	0.4025	0.7030	0.5885	0.7281	0.5164
TiRex	0.3007	0.4268	0.6831	0.5731	0.7158	0.4885
Chronos-Bolt	0.2652	0.3889	0.7328	0.6200	0.8076	0.5743

FEV evaluates heterogeneous variate roles, TIME emphasizes task-specific forecasting configurations, and GIFT-Eval spans diverse temporal resolutions and forecast lengths. Reporting the metrics separately avoids combining point and distributional accuracy into an arbitrary composite score.

C.2 FEV Task Types

The five groups partition all FEV configurations. Univariate and multivariate groups have no dynamic covariates; the remaining groups distinguish past-only inputs, known-future inputs, and their combination. Covariate tasks may contain multiple target variates. Each group uses the same geometric aggregation as the overall benchmark, rather than an arithmetic average of the displayed skill values.

Table 6: FEV MASE skill by task type. Groups are mutually exclusive and include all 100 configurations; higher is better. Best and second-best values are marked within each row.

Task type	n	Timer-M1	TimesFM-3	Chronos-2	TiRex-2	Toto 2.0
Univariate	32	0.2968	**0.2998**	0.2799	0.2629	0.2653
Multivariate	26	0.4266	0.4252	0.4017	0.3889	**0.4366**
Past-only	12	**0.4256**	0.3933	0.3655	0.3969	0.3426
Known-future	18	0.4481	**0.4549**	0.4323	0.3979	0.3332
Known + past	12	0.2988	0.2923	**0.3029**	0.2465	0.1710

Table 7: FEV SQL skill by task type. Groups are mutually exclusive and include all 100 configurations; higher is better. Best and second-best values are marked within each row.

Task type	n	Timer-M1	TimesFM-3	Chronos-2	TiRex-2	Toto 2.0
Univariate	32	0.3828	**0.3847**	0.3703	0.3538	0.3563
Multivariate	26	0.5954	0.5953	0.5795	0.5668	**0.6033**
Past-only	12	**0.4668**	0.4382	0.4239	0.4387	0.3950
Known-future	18	0.5363	**0.5458**	0.5257	0.4971	0.4344
Known + past	12	**0.4295**	0.4171	0.4254	0.3765	0.3022

The breakdown retains both point and probabilistic scores to distinguish temporal forecasting from the use of additional cross-variate information. In particular, separating past-only and known-future inputs distinguishes inference from related history from conditioning on observations available over the forecast interval. This organization follows the variate roles used to construct forecasting episodes, while keeping the evaluation tasks fixed across models. Overall FEV skill is recomputed from all configuration-level errors, not from an unweighted mean of the five group scores, because group sizes differ.

C.3 PRIMITIVE GROUPS

Tables 8–10 expand the groups displayed in Figure 8. They retain every record within each displayed group and all available compared models. These groups are a selected view of the diagnostic suite, not its exhaustive inventory; no overall rank over the entire suite is inferred from them. Window lengths, observation conditions, and metric definitions are specified in Appendix B.5. The count n denotes scoring records, which may include multiple views of the same generated task.

Table 8: Temporal extrapolation groups corresponding to Figure 8. Mean squared error is averaged over the complete records of each displayed group; lower is better.

Group	n	Timer-M1	TimesFM-3	Chronos-2	TiRex-2	Toto 2.0
Shift	6	**6.4278**	7.4830	6.8490	6.9092	6.4418
Periodic	13	**0.0271**	0.6803	0.0919	0.0735	0.2300
Trend	10	0.1312	0.0299	0.4522	0.4512	**0.0083**

Table 9: Relational forecasting groups corresponding to Figure 8. Mean squared error is averaged over the complete records of each displayed group; lower is better.

Group	n	Timer-M1	TimesFM-3	Chronos-2	TiRex-2	Toto 2.0
Sign	2	**0.0021**	0.0036	0.0077	0.0100	0.0073
Cointegration	2	**0.2477**	0.3953	2.2627	1.3930	4.7848
Promotion	2	**0.3008**	0.3042	1.0424	0.5567	–
Lag	2	0.0012	**0.0005**	0.0185	0.0074	0.0049
Phase	2	0.000053	**0.000043**	0.0002	0.0087	0.0010
Scale	2	**0.0688**	0.0955	0.8690	2.2971	4.7189
Shift	2	**0.1060**	0.2068	0.2593	0.3129	0.4098

Table 10: Input reliability groups corresponding to Figure 8. Mean squared error is averaged over the complete records of each displayed group; lower is better.

Group	n	Timer-M1	TimesFM-3	Chronos-2	TiRex-2	Toto 2.0
Irrelevant context	7	0.3681	0.2588	0.2554	0.2585	**0.2451**
Irrelevant targets	3	**0.0035**	–	0.0167	0.0633	0.0055
Missing event inputs	7	5.7917	5.7594	3.9130	**2.2120**	–
Target corruption	21	12.0938	2.6081	23.9802	**0.2260**	3.8370
Target missingness	21	0.1957	**0.1477**	0.7231	0.2184	0.3595
Variate order	7	**0.0034**	–	0.0221	0.0623	0.0055

Dashes denote unsupported inputs or unavailable squared-error results, not failed forecasts assigned an artificial penalty. TimesFM-3 uses the fixed configuration for these controlled tests; public-benchmark scores follow their own evaluation settings. Selected examples in Appendix D illustrate the observed temporal patterns and relations without changing the numerical groups above.

C.4 Ablation Results

Gate controls are independently trained with fixed residual multipliers, retaining variate attention and the same residual order. Fixed 0.202 uses the unrounded stable-reference mean, approximately 0.201676, while Fixed 1.0 uses the full variate-attention update. Adaptive gates are learned separately at each layer and remain fixed during inference. These comparisons are not inference-only interventions on a common checkpoint.

Table 11: Gate ablation on FEV overall and five task types. Both metrics report skill scores. Red bold and underline indicate the best and second-best results within each column and metric. Adaptive gates achieve the highest overall skill, with the clearest advantage in known-future forecasting.

Gate	Overall	Univariate	Multivariate	Past-only	Known-future	Known + past
			MASE Skill ↑			
Adaptive	**0.3771**	0.2968	0.4266	0.4256	**0.4481**	0.2988
Fixed 0.202	0.3714	**0.2969**	**0.4299**	**0.4266**	0.4084	0.3081
Fixed 1.0	0.3680	0.2963	0.4260	0.4238	0.3979	**0.3104**
			SQL Skill ↑			
Adaptive	**0.4887**	**0.3828**	0.5954	**0.4668**	**0.5363**	0.4295
Fixed 0.202	0.4836	0.3826	**0.5981**	0.4651	0.5002	0.4406
Fixed 1.0	0.4814	0.3819	0.5956	0.4659	0.4920	**0.4426**

Table 12: Data-construction ablations across three benchmarks. Recipe names refer to the source-generation routes under comparison; shared benchmark-associated training data are unchanged across variants.

Recipe	FEV MASE ↑	FEV SQL ↑	TIME MASE ↓	TIME CRPS ↓	GIFT MASE ↓	GIFT CRPS ↓
Real Only	0.3196	0.4431	0.6579	0.5526	0.6998	0.4850
Synthetic Only	0.3568	0.4710	0.6438	0.5456	0.6884	0.4741
No-Joint	0.3578	0.4725	0.6379	**0.5359**	0.6819	0.4722
Full Recipe	**0.3771**	**0.4887**	**0.6372**	0.5363	**0.6806**	**0.4688**

Real Only and Synthetic Only restrict the source-generation routes under comparison; No-Joint retains both source types but removes corpus-derived and mixed-source joint generation. Shared GIFT-Eval and TIME training collections remain unchanged across the four variants. Consequently, these names describe the controlled routes rather than the exclusive provenance of all training observations. The recipe comparisons assess inclusion of generation routes and do not isolate each individual operator.

D PRIMITIVE EVALUATION SHOWCASE

The following pages present 24 illustrative forecasts, eight for each capability. Examples cover distinct temporal patterns, cross-variate relations, and observation conditions, with an emphasis on mechanisms discussed in the main text. They are selected examples rather than a random sample for estimating overall success rates. All model curves within a panel use the same target and forecast origin. The dashed boundary separates history from the forecast interval; shading denotes Timer-M1's marginal 10–90% prediction quantiles, not confidence intervals for its mean prediction.

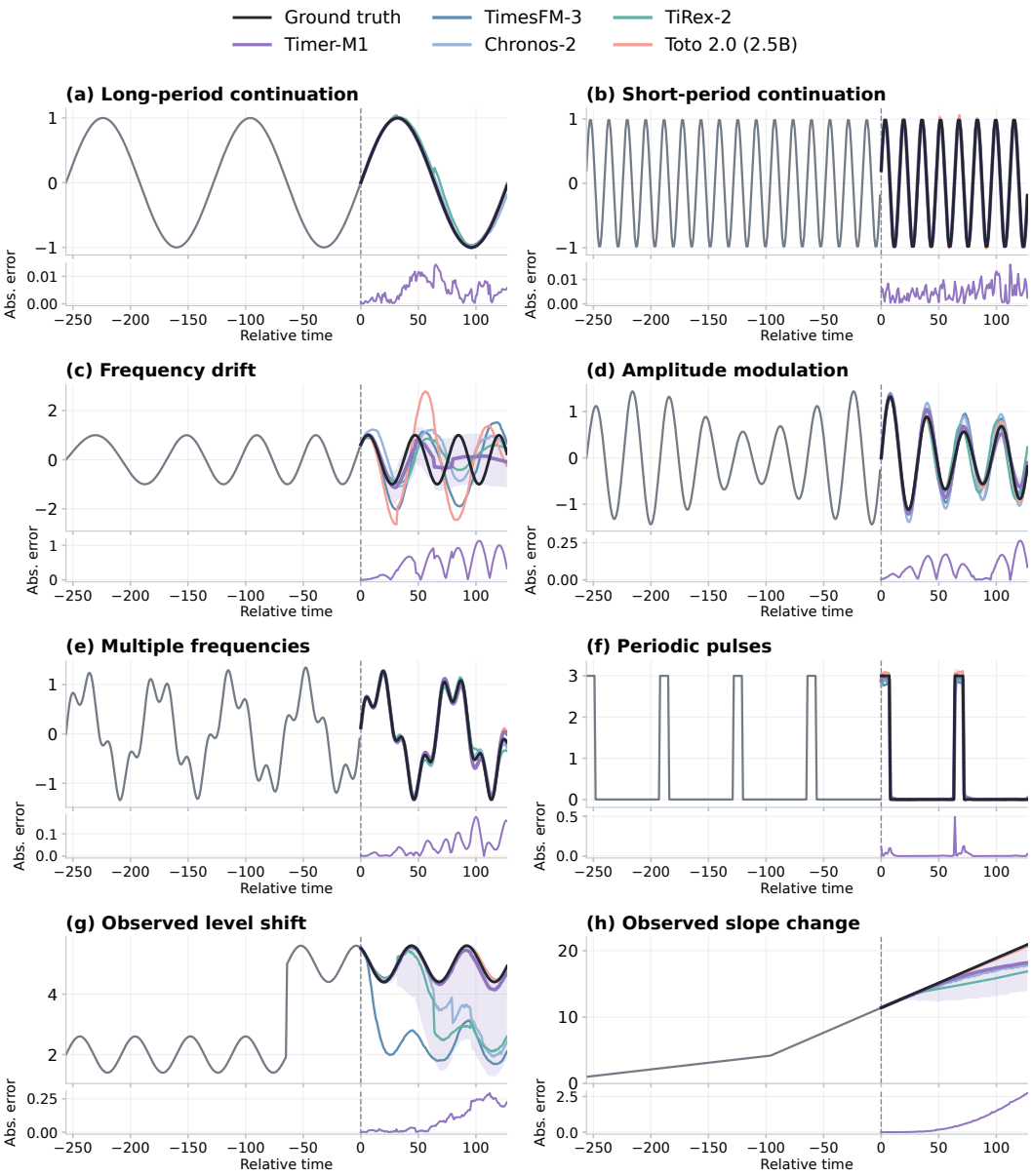

Figure 11: Temporal extrapolation across periodicity, modulation, frequency drift, and observed regime changes. Lower strips show Timer-M1's absolute forecast error. Each panel uses its own value scale.

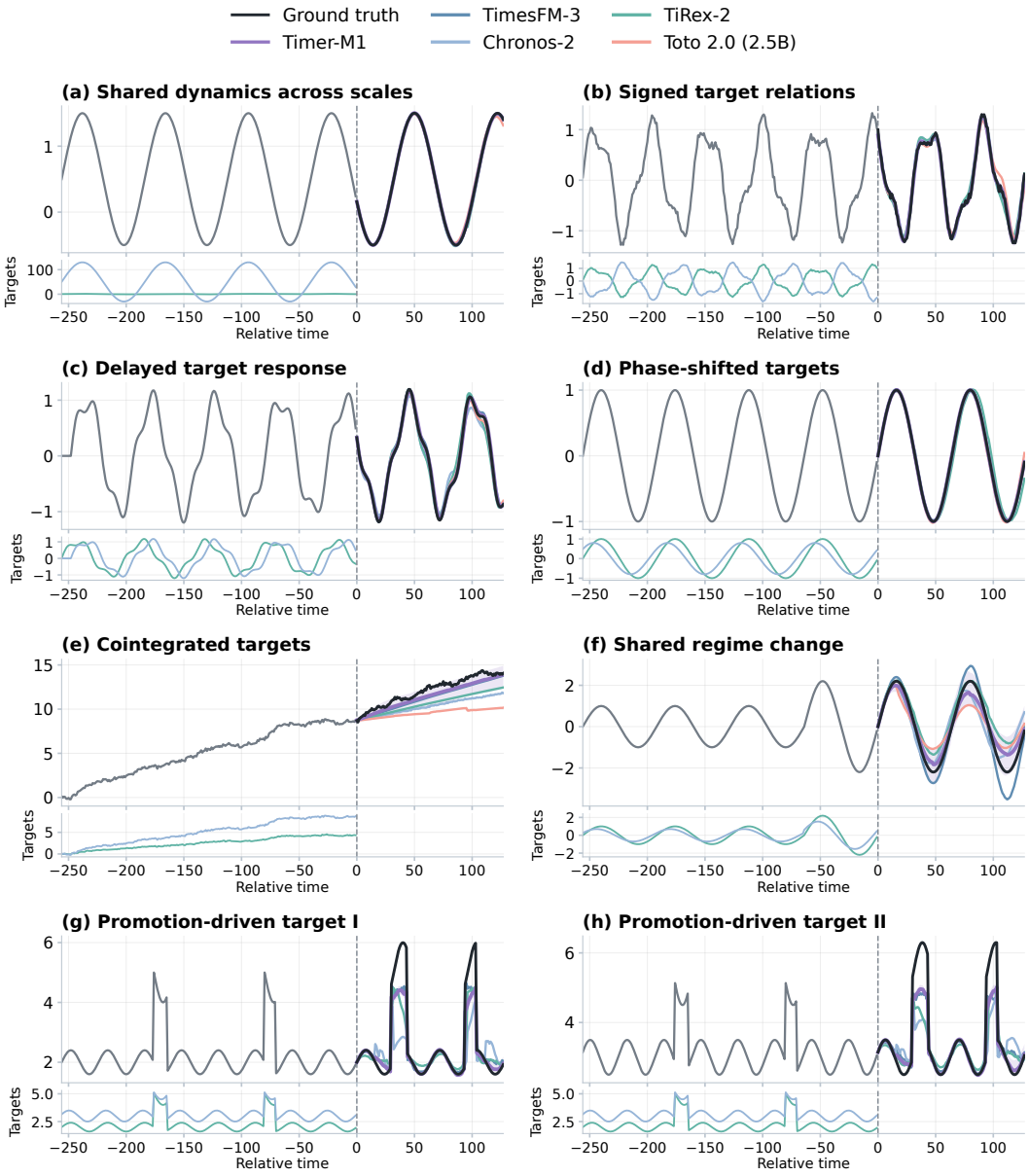

Figure 12: Relational forecasting with shared, signed, delayed, phase-shifted, cointegrated, and event-driven dynamics. Lower strips show available related inputs or target histories; they are not additional forecast targets unless specified by the task.

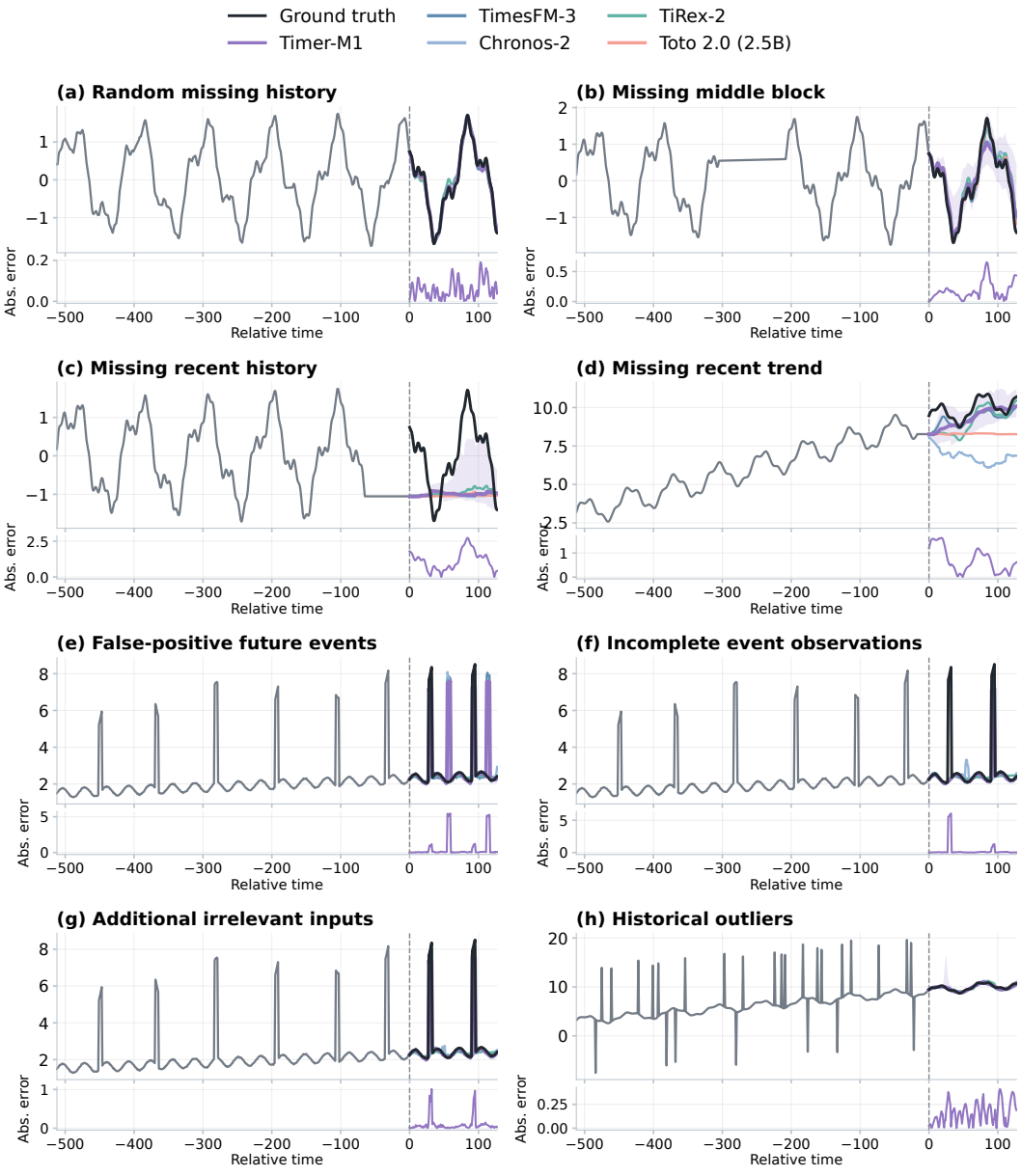

Figure 13: Input reliability under missing histories, imperfect event information, irrelevant inputs, and historical outliers. Lower strips show Timer-M1's absolute forecast error.