MORPHEUS-2.0: MULTISCALE PROBABILISTIC FORECASTING WITH DUAL-AXIS ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

A universal forecaster is trained once and applied everywhere, so it must serve tasks whose schemas, forecast horizons, and temporal resolutions all differ. The number of targets and covariates changes from dataset to dataset, horizons vary by orders of magnitude, and sampling rates range from seconds to years. Existing time-series foundation models address these only in part, since they isolate each series or visit the two grid relations in separate blocks, and they fix the horizon and resolution before training begins. We introduce **MORPHEUS-2.0**, a compact encoder-only foundation model that represents targets and covariates on a single variable–time grid. Its dual-axis attention runs both relations side by side in every layer, one across variables at a timestep and one across time within a variable; a learned gate weights the two per layer, and an axis-aware rotary encoding preserves both grid coordinates. The forecast length is sampled during training, so one set of weights spans short to long horizons at no additional inference cost. **MORPHEUS-MULTISCALE** extends the model to several temporal resolutions at once, weighting them according to the series being forecast. Across GIFT-Eval, which spans several domains and frequencies, the two releases give complementary operating points: the MORPHEUS-2.0 attains 0.689 relative MASE with 81M parameters, while the larger MORPHEUS-MULTISCALE attains **0.653**, the lowest relative MASE of any published system on the benchmark.

1 INTRODUCTION

Forecasting supports decisions in energy, retail, finance, climate, and cloud operations, yet conventional systems are tuned to a particular dataset, frequency, and horizon, which makes broad deployment expensive to maintain. Time-series foundation models (TSFMs) offer a different paradigm: pretraining on heterogeneous series lets one model transfer across domains with little or no task-specific optimization (Ansari et al., 2024; Das et al., 2024; Woo et al., 2024; Cohen et al., 2025). The central problem then becomes building a single model whose representation and objective remain effective across that heterogeneity.

That heterogeneity is structural, and it runs along three axes. A forecasting example forms a two-dimensional variable–time grid, and the first axis is its *schema*: how many rows that grid has, its *schema width*, and what each row is, whether a target to be predicted, a past-only covariate, or a covariate known into the future (Ansari et al., 2025; Khwaja et al., 2026; Podest et al., 2026). Both change from task to task. Channel-independent processing sidesteps this but cannot exploit cross-variable structure, while unrestricted attention over the grid pays quadratic cost on both axes and provides no inductive bias to separate relations across variables and dynamics within a variable.

The second and third axes are both temporal, the forecast horizon and the patch resolution. One is the forecast horizon: training at a single fixed length over-specializes the model, since short output blocks force iterative rollout by feeding forecast back into context, while very wide blocks make every training prediction harder. The other is temporal resolution. We group each variable into consecutive blocks of s timesteps, a *patch*, and treat each as one token; s is the *patch width*. Narrow patches resolve brief events, wide ones reach further back and smooth what they cover, so no single choice suits data sampled from seconds to years. Section 7 examines both effects.

We introduce **MORPHEUS-2.0**, an encoder-only TSFM that casts forecasting as masked prediction over an interleaved variable–time grid, with covariates and targets sharing one flexible-schema

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.

representation and masks enforcing the information available at the forecast origin. In every layer, dual-axis attention runs two pathways, one across variables at the same timestep and one across time within the same variable, fused by a learned gate. This yields $O(TN^2 + T^2N)$ permitted pairs for T times and N variables instead of $O(T^2N^2)$, executed with block-sparse FlexAttention (Dong et al., 2024), while axis-aware two-dimensional RoPE preserves the grid coordinates. Prediction-length randomization samples the forecast length for each training batch, so one parameter set covers short and long requests alike, with no horizon-specific heads and no added inference cost.

We further introduce **MORPHEUS-MULTISCALE**, which represents each series at several complementary patch widths at once, maps the observed context and requested horizon to a weighting over those widths, and merges their aligned marginal quantiles. Across the 97 configurations of GIFT-Eval (Aksu et al., 2024), the 81M-parameter MORPHEUS-2.0 reaches 0.689 relative mean absolute scaled error (MASE), competitive with substantially larger systems, and MORPHEUS-MULTISCALE reaches 0.653, the lowest relative MASE of any publicly reported system (Table 1).

Our main contributions are:

- A novel two-dimensional variable–time representation with axis-factorized sparse attention, gated cross-axis fusion, and axis-aware two-dimensional RoPE.
- Prediction-length randomization for training one probabilistic model across forecast horizons without additional inference cost.
- Two released models at complementary operating points: a compact MORPHEUS-2.0, and MORPHEUS-MULTISCALE, which weights several patch widths per instance to reach the lowest published relative MASE on GIFT-Eval.

2 RELATED WORK

Statistical baselines under automatic order selection remain competitive (Hyndman & Khandakar, 2008), and the M4 competition was won by a hybrid of exponential smoothing and a recurrent network (Makridakis et al., 2020). Deep models displaced them by training one network across many series (Salinas et al., 2020; Oreshkin et al., 2020), with the Temporal Fusion Transformer adding covariate pathways (Lim et al., 2021), until LTSF-Linear matched the long-horizon transformers that followed with a one-layer linear map (Zeng et al., 2023).

Foundation models remove the per-dataset fitting that the models above require: one set of pretrained weights forecasts unseen series, which leaves open how to represent a series generally enough for that to work. Chronos discretizes scaled values into token sequences, TimesFM uses a patched decoder with direct output patches, and Moirai broadened this to masked, any-variate probabilistic forecasting (Ansari et al., 2024; Das et al., 2024; Woo et al., 2024); TiRex instead adopts an xLSTM backbone and Sundial replaces discrete tokenization with flow matching (Auer et al., 2025; Liu et al., 2025). Recent systems model variables jointly, with Chronos-2 and TiRex-2 supporting multivariate targets alongside past-only and known-future covariates and Toto 2.0 keeping a multivariate time/variate backbone (Ansari et al., 2025; Podest et al., 2026; Cohen et al., 2025; Khwaja et al., 2026). MORPHEUS-2.0 follows Morpheus (Patil et al., 2025) and these systems in adopting the covariate-aware multivariate setting.

How that setting is represented varies widely. Channel-independent patching treats each series on its own, robust but blind to contemporaneous signal (Nie et al., 2023); iTransformer makes a whole variate one token, and Crossformer keeps both axes but visits them in stages (Liu et al., 2024a; Zhang & Yan, 2023), instances of the axial factorization template (Ho et al., 2019). Among foundation models, Moirai and Morpheus flatten variable–time patches into one sequence, Chronos-2 patches series separately and attends within a related group, and TiRex-2 keeps a variate-by-patch tensor through alternating mixers (Woo et al., 2024; Patil et al., 2025; Ansari et al., 2025; Podest et al., 2026). Several visit the two axes in turn; MORPHEUS-2.0 evaluates both in every layer and learns their balance at each depth.

A second axis of variation is how the horizon and the temporal scale are supervised. MQ-RNN established direct multi-horizon quantile heads (Wen et al., 2017), and TimesFM moved from autoregressive output patches to filling a masked horizon in one pass (Das et al., 2024; Jain & Sen, 2026). Training also varies the span, through context–prediction splits, output patch counts, or variable-length masked spans (Woo et al., 2024; Ansari et al., 2025; Khwaja et al., 2026), and across

scales Moirai learns patch-size-specific projections, Tiny Time Mixers pairs adaptive patching with resolution sampling, and N-HiTS uses multi-rate sampling with hierarchical interpolation (Woo et al., 2024; Ekambaram et al., 2024; Challu et al., 2023). Data is the other limiting resource: public corpora have grown largely by restacking the pool Monash standardized (Godahewa et al., 2021; Woo et al., 2024; Liu et al., 2024b; Shi et al., 2025; Liu et al., 2025), so the independent sources beneath them are fewer than their totals suggest, and generated series are now a standard component of pretraining mixtures (Ansari et al., 2024; Das et al., 2024). MORPHEUS-2.0 follows this practice with MySTiC (Section 5.1).

3 PROBLEM FORMULATION

A forecast is issued at time T for the next H timesteps. Each example contains N variables on a common time axis, and each variable is one of three types: $N_y \geq 1$ *targets*, whose future values are predicted; N_p *past-only covariates*, observed up to T; and N_f *known-future covariates*, such as calendar features or planned controls, available up to $T+H$. Every value carries a binary observation mask that separates missing entries from observed zeros. We write $\mathcal{C}_T$ for everything the model may use when the forecast is issued: the history of every variable up to T and the known-future covariates up to $T + H$. The schema (N_y, N_p, N_f), the context length, the horizon, and the missingness pattern all vary across examples, and we refer to each variable as a *row* of the grid representation of Section 4.1.

For each target j and lead time h, a single shared model predicts quantiles of $y_{j,T+h}$ given $\mathcal{C}_T$: the τ-quantile is the value that $y_{j,T+h}$ falls at or below with probability τ, and the model predicts it as $\widehat{Q}_{j,h}(\tau)$ for every level τ in a fixed grid $\mathcal{Q}$ (Appendix C3). The median $\widehat{Q}_{j,h}(0.5)$ is the point forecast, and the remaining levels describe uncertainty around it. Formally, we seek a single parameter vector θ whose map $f_\theta : (\mathcal{C}_T, H) \mapsto \{\widehat{Q}_{j,h}(\tau)\}$ minimizes the expected quantile loss (Section 4.2) across schemas, horizons, and sampling frequencies.

4 METHODOLOGY

Figure 1 summarizes the architecture. MORPHEUS-2.0 is an encoder-only transformer trained under one pinball objective (Koenker & Bassett, 1978) and following the masked-encoder formulation of Moirai and Chronos-2 (Woo et al., 2024; Ansari et al., 2025).

4.1 FLEXIBLE-SCHEMA VARIABLE–TIME REPRESENTATION

Let s denote the patch width and $n \in \{1, \ldots, N\}$ index the concatenated past-covariate, known-future-covariate, and target rows. For temporal patch k, MORPHEUS-2.0 embeds the value vector $\mathbf{v}_{k,n} \in \mathbb{R}^s$ concatenated with the matching slice $\mathbf{m}_{k,n} \in \{0,1\}^s$ of the observation masks of Section 3, giving one token $\mathbf{z}_{k,n} = \mathrm{PatchEmbed}([\mathbf{v}_{k,n}; \mathbf{m}_{k,n}])$. Tokens are flattened in time-major order, so all variables at a temporal patch are contiguous, while explicit time, variable, and sample identifiers retain the grid structure, and shared patch and output projections permit N to vary without dataset-specific heads. A fixed token budget P bounds this flexibility: at most $\lfloor P/N \rfloor$ temporal patches are available per row. Most practical tasks have $N \leq 50$, and the model accepts any N, with fewer temporal patches per row as N grows.

4.2 MASKED-HORIZON QUANTILE FORECASTING

MORPHEUS-2.0 places context and forecast positions in one encoder sequence. A token j is an admissible key exactly when n_j is a known-future row or $t_j \leq T$: history is visible bidirectionally, future targets and past-only covariates become masked placeholders that remain queries, and known-future covariates stay visible past T because the task supplies them. Both neighborhoods of Section 4.3 are intersected with that key set and the loss is evaluated only at future target positions, so each target query attends to everything available at the forecast origin without reaching another target's future value along either axis. At each masked target position the output head directly predicts marginal quantiles $\{\widehat{Q}_{j,h}(\tau) : \tau \in \mathcal{Q}\}$, trained with the pinball loss $\rho_\tau(u) = \max\{\tau u, (\tau - 1)u\}$ on $u = y - \widehat{Q}(\tau)$ (Koenker & Bassett, 1978), averaged over observed future targets and summed

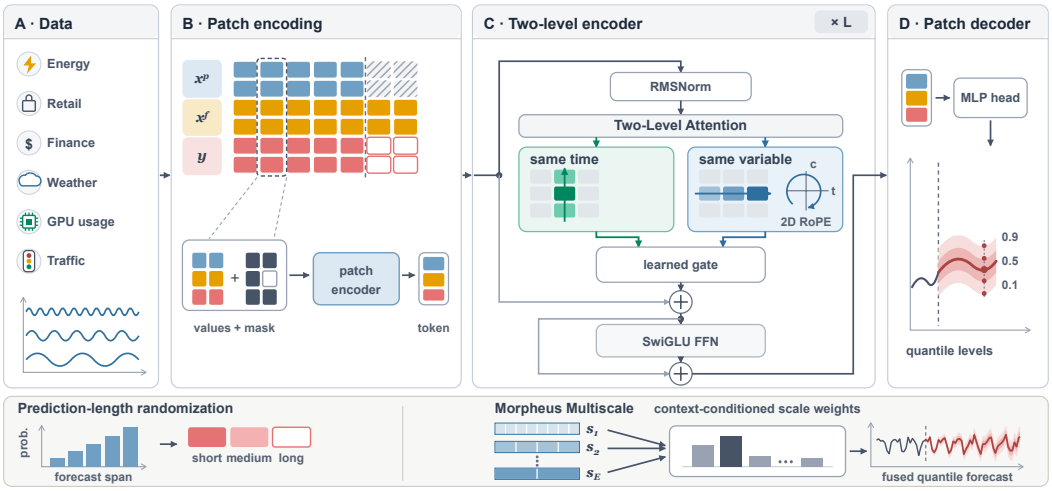


Figure 1: Overview of MORPHEUS-2.0. **(A)** Training data span many domains, frequencies, and schemas. **(B)** Covariates and targets form one variable–time grid; each patch is embedded with its observation mask, so one time patch across all rows becomes one token, and future targets remain query-only. **(C)** Every block applies two parallel attention pathways, same-time across variables and same-variable across time with axis-aware two-dimensional RoPE, combined by a learned scalar gate. **(D)** The patch decoder projects each token to the native quantile grid, with one prediction length sampled per batch. MORPHEUS-MULTISCALE runs this pipeline at several patch resolutions and fuses them (Section 4.6).

over levels. The whole native block is emitted in one forward pass, and longer requests append the predicted medians to the context and repeat; this direct multi-horizon quantile interface (Wen et al., 2017; Ansari et al., 2025; Khwaja et al., 2026) is what lets different patch widths be aligned and combined.

4.3 DUAL-AXIS ATTENTION

Let token i have temporal index t_i and variable index n_i, the row index of Section 4.1. Its intra-time and inter-time neighborhoods are $\mathcal{I}(i) = \{j : t_j = t_i\}$ and $\mathcal{T}(i) = \{j : n_j = n_i\}$, with padding and cross-sample pairs removed. Independent projections attend over each, giving $\mathbf{a}_i^{\mathcal{I}}$ and $\mathbf{a}_i^{\mathcal{T}}$, which layer ℓ fuses through output projections $W^{\mathcal{I}}$ and $W^{\mathcal{T}}$ as

$$\mathbf{u}_i = (1 - g_\ell) W^{\mathcal{I}} \mathbf{a}_i^{\mathcal{I}} + g_\ell W^{\mathcal{T}} \mathbf{a}_i^{\mathcal{T}}, \qquad g_\ell = \mathrm{sigmoid}(\gamma_\ell), \tag{1}$$

with γ_ℓ one unconstrained trainable scalar per layer, learned from the forecasting loss alone (Appendix A1). A fixed $g_\ell = 1/2$ would hold one balance at every depth; nine learned scalars let each block set its own balance (Section 7.1).

Evaluating both axes inside one layer is the property we import from vision transformers, whose patch tokens keep their row–column coordinates so any layer can relate two patches along either axis (Dosovitskiy et al., 2021). A forecasting example has the same structure, so we keep simultaneous access to both axes while replacing attention over all $(TN)^2$ pairs with attention over $\mathcal{I}(i) \cup \mathcal{T}(i)$. Recent forecasting backbones instead alternate the two relations across blocks (Ho et al., 2019; Zhang & Yan, 2023; Ansari et al., 2025; Khwaja et al., 2026; Podest et al., 2026), updating a token by one relation at a time, so a dependence coupling the two axes is never formed inside a block and survives only if the intermediate representation carries the first relation forward to the block supplying the second. This matters most for covariate-conditioned forecasting, where a target must associate a covariate observed now with the temporal pattern carried by that covariate's history; Section 7.1 isolates the operator at matched capacity. Restricting attention to pairs that share a coordinate also reduces cost, since the two neighborhoods admit $TN^2 + NT^2$ pairs compared with $(TN)^2$ under unrestricted attention, and the difference grows with the size of the grid. Block-sparse FlexAttention realizes the retained pairs directly instead of masking a dense score matrix, so the discarded ones are never materialized.

4.4 POSITIONAL ENCODING

Attention ignores token order, so each token must carry its grid position. RoPE (Su et al., 2024; Heo et al., 2024) makes the attention score depend only on the offset between two tokens, but applied to the flattened index $p = tN + n$ it ties time offsets to the schema width N, and applied to time alone it tells the intra-time pathway nothing about which variables it compares. We rotate the r-th rotary pair of each query and key by

$$\phi_r(t, n) = t\,\omega_r + \alpha\, n\,\omega_r, \tag{2}$$

where t is the time index, n the variable index, ω_r the standard RoPE frequencies, and $\alpha = 0.04$. The intra-time pathway compares tokens with equal t and so sees only Δn; the inter-time pathway compares tokens with equal n and so sees only Δt. The two coordinates therefore cannot alias (Appendix A2). α and the rotary base are fixed by a sweep on held-out synthetic data; the inter-time pathway also adds a learned relative-position bias over time.

4.5 PREDICTION-LENGTH RANDOMIZATION

The forecast horizon is set by the task, and requested spans run from minutes of grid load to quarters of financial planning. Training at one fixed prediction length optimizes exactly that span at every step: shorter requests are never supervised on their own, and longer ones must be reached by iterative rollout, which accumulates error. Narrowing the block to fix the first problem only worsens the second.

Each training batch samples one prediction-token budget M, the number of prediction patches, and derives a per-example horizon from it: for an example with N_i interleaved rows and patch width s,

$$M \in \{m_1, \ldots, m_K\}, \quad \Pr(M = m_k) = \pi_k, \qquad H_i(M) = s \max(1, \lfloor M/N_i \rfloor), \tag{3}$$

and training optimizes $\mathbb{E}_M \mathbb{E}_i[\mathcal{L}_{H_i(M)}]$. Sampling a common token budget preserves a uniform layout within the batch while exposing the shared parameters to several context–horizon partitions. Related models vary context–prediction boundaries, output-patch counts, or contiguous masked spans (Woo et al., 2024; Ansari et al., 2025; Khwaja et al., 2026; Podest et al., 2026). Contiguous patch masking hides an interior span reconstructed from context on both sides, an infilling objective; prediction-length randomization varies the length of a terminal region with no right context, so every sampled task is a forecast. MORPHEUS-2.0 also randomizes at a different level from these schemes. Because every example is laid out as an interleaved variable–time canvas under a shared token budget, one draw of M yields a timestep horizon H_i that differs across examples with different numbers of rows (Eq. 3), so a single parameter set is trained across both a range of horizons and a range of schema widths. At inference a requested horizon is served by masking that many future positions.

4.6 MORPHEUS-MULTISCALE: CONTEXT-CONDITIONED SCALE FUSION

A MORPHEUS-2.0 encoder with patch width s_e produces marginal quantiles $\widehat{Q}_e(\tau \mid \mathcal{C}_T)$ for every target and lead time, so encoders trained at different widths emit aligned outputs that can be combined directly. MORPHEUS-MULTISCALE combines $E \geq 2$ such encoders, each trained at its own patch width: narrow widths resolve local detail, wide widths longer temporal structure. For each forecast, a *mixing head* A_ϕ takes the context $\mathcal{C}_T$ and the requested horizon H and maps them to weights over the E encoders, $\mathbf{w}_\phi(\mathcal{C}_T, H) = \mathrm{softmax}(A_\phi(\mathcal{C}_T, H))$, which are nonnegative and sum to one. The forecast is their weighted average,

$$\widehat{Q}_{\mathrm{multiscale}}(\tau \mid \mathcal{C}_T, H) = \sum_{e=1}^{E} w_{\phi,e}(\mathcal{C}_T, H)\, \widehat{Q}_e(\tau \mid \mathcal{C}_T), \qquad \tau \in \mathcal{Q}. \tag{4}$$

The same weights apply to every target, lead time, and each of the 21 levels of $\mathcal{Q}$ (Appendix C3), so the combination averages whole quantile functions, known as Vincentization (Lichtendahl et al., 2013), rather than mixing predictive distributions. Since the pinball loss is convex in the prediction, Jensen's inequality gives $\rho_\tau(y - \sum_e w_e \widehat{Q}_e(\tau)) \leq \sum_e w_e\, \rho_\tau(y - \widehat{Q}_e(\tau))$ at every level and hence for their average, so the combination is never worse than the weighted average of its components under the benchmark's own metric.

The mixing head is a single MORPHEUS-2.0 layer with a narrower hidden size, followed by an attention-pooling step that summarizes the context into a fixed-length vector. Training is sequential:

the encoders are first trained under the objective of Section 4.5, and the mixing head is then trained on the errors the encoders make (Appendix C2).

5 Experimental Setup

5.1 Pretraining Data

MORPHEUS-2.0 is pretrained on a mixture of real and synthetic series. The real portion is the public GIFT-Eval pretraining corpus, roughly 230 billion observations across diverse domains and frequencies (Aksu et al., 2024); we train on all of it, assigning each domain–frequency combination its own sampling weight, since a few very large sources would otherwise drown out the rest.

Real collections alone leave several regimes underrepresented, in particular wide multivariate schemas, informative known-future covariates, and controlled degrees of nonstationarity. We therefore construct the **Multivariate Synthetic Time-series Collection (MySTiC)**, a generated corpus supplying those regimes through three subsets, the first of which extends the generator of Morpheus (Patil et al., 2025): composed Gaussian-process kernels with factor and lead–lag coupling, giving correlated panels at every schema width; covariates attached to targets through known response functions, so covariate-conditioned supervision has its dependence fixed by construction; and emulated signals that reproduce varied temporal structure (Appendix B2).

5.2 Training Details

All reported models instantiate one encoder specification and differ only in patch width and prediction-token budget (Appendix A3). We train each with AdamW (Loshchilov & Hutter, 2019) and a one-cycle learning-rate schedule (Smith, 2018), using standard multi-GPU data parallelism and bfloat16 arithmetic, with training budgets set by realized patch-token exposure per trainable parameter (Hoffmann et al., 2022). Hyperparameters are tuned on held-out synthetic multivariate data.

The four-stage data curriculum progressively increases distributional complexity and realism, moving from elementary synthetic dynamics through structurally richer synthetic processes to a final stage dominated by real observations, so that reusable temporal primitives are acquired before the model must contend with the noise and irregularity of observed series. MORPHEUS-MULTISCALE adds one further stage, in which the mixing head of Section 4.6 is fitted on top of the trained encoders. Stage composition appears in Appendix B3, and packing, optimization, and hardware in Appendix A4 and Appendix A3.

5.3 Benchmark, Metrics, and Protocol

GIFT-Eval (Aksu et al., 2024) contains 23 datasets, more than 144,000 series, and 177 million observations across seven domains, with ten frequencies spanning seconds to years, univariate and multivariate data, and multiple horizons, giving 97 configurations with seasonal, trending, intermittent, bursty, regime-changing, and cross-variable dynamics. We evaluate every test item under the rolling-origin protocol and compare against publicly reported results obtained under the same protocol.

We report two headline metrics, with MSE, MAPE, sMAPE, and ND in Appendix E2. Point accuracy is the mean absolute scaled error (MASE): the absolute error of the median forecast divided by the in-sample error of a seasonal naive predictor on the same item, which makes it scale free across series of very different magnitude (Hyndman & Koehler, 2006). Probabilistic accuracy is the continuous ranked probability score (CRPS), which GIFT-Eval approximates by the mean weighted quantile loss over the 9 levels $\{0.1, \ldots, 0.9\}$, normalized by total absolute label mass (Aksu et al., 2024; Alexandrov et al., 2019). Both metrics are aggregated within each configuration, then reported as a geometric mean (GM) over the 97 configurations relative to Seasonal Naive, which therefore scores 1.0; longer requests use iterative rollout (Appendix C3).

Table 1: GIFT-Eval results over 97 configurations. Rel. MASE scores the median forecast and Rel. CRPS the full quantile forecast, both GM relative to Seasonal Naive; lower is better. Setup records whether benchmark training data are used; parameter counts are taken from the official benchmark.

Model	Params	Setup	Rel. MASE ↓	Rel. CRPS ↓
Seasonal Naive	–	Reference	1.000	1.000
Moirai-2.0-R-Small	11.4M	Pretrained	0.728	0.516
TimesFM-2.5	200M	Zero-shot	0.705	0.490
Chronos-2	120M	Pretrained	0.698	0.485
MORPHEUS-2.0[1]	81.4M	Pretrained	0.689	0.522
Toto-2.0-2.5B-FT	2.5B	Fine-tuned	0.679	<u>0.463</u>
TiRex-2-Pretrained	82.5M	Pretrained	0.678	0.467
Toto-2.0-FnF	N/A	Pretrained	0.676	<u>0.463</u>
TimesFM-3	330M	Zero-shot	0.667	**0.456**
Falcon-2.0	585M	Pretrained	<u>0.666</u>	0.486
MORPHEUS-MULTISCALE[2]	247M	Pretrained	**0.653**	0.476

[1] $\sigma_{\mathrm{MASE}} = 0.0014$, $\sigma_{\mathrm{CRPS}} = 0.0010$; [2] $\sigma_{\mathrm{MASE}} = 0.0011$, $\sigma_{\mathrm{CRPS}} = 0.0016$.

Table 2: Attention-operator comparison. All metrics are relative; lower is better.

Attention operator	MASE	CRPS	Multivariate MASE
Full attention over the grid	0.763	0.570	0.779
Dual-axis attention, fixed gate	0.729	0.553	0.753
Dual-axis attention, learned gate	**0.689**	**0.522**	**0.708**

6 MAIN RESULTS

Table 1 compares both releases with public GIFT-Eval submissions scored under the same protocol; the full leaderboard and a per-domain breakdown are in Appendix E1 and Appendix E3. MORPHEUS-MULTISCALE combines three encoders of about 82M parameters each, at patch widths 2, 4, and 8, under a small mixing head (Appendix A3). It reaches the lowest relative MASE of any published system. Both of our rows are means of five seeds, and the seed standard deviations are well below the margins discussed here.

Both releases outrank larger systems trained under the same setup: MORPHEUS-MULTISCALE improves on Falcon-2.0 by 2.0% relative MASE with 42% of its parameters, and MORPHEUS-2.0 is more accurate than Chronos-2 with a third fewer parameters. Combining widths improves on the standalone model by 5.2% in MASE and 8.8% in CRPS, in every horizon group, and puts MORPHEUS-MULTISCALE ahead of Chronos-2 and Falcon-2.0 on both metrics. On CRPS it is 4.4% behind the best system, a gap in predictive spread rather than location (Appendix E4). By horizon, MORPHEUS-MULTISCALE is the most accurate system on medium and long horizons and is less accurate than TimesFM-3 at short horizons (Table 13). By domain, among the systems compared, it is the most accurate on Energy and Transport, which together account for nearly half of the configurations (Table 15).

7 ABLATIONS AND ANALYSIS

We examine the four design choices of Sections 4.3–4.6 together with the pretraining recipe of Section 5.1. Each comparison is a controlled study that varies only the factor under test, and significance statements use 95% paired bootstrap intervals resampled over configurations (Efron, 1979).

7.1 DUAL-AXIS ATTENTION

Because the grid is already serialized into one token sequence, unrestricted attention over all pairs is available as a baseline. Table 2 compares three nine-layer encoders matched in width, patch size, and every other architectural and training setting: full attention, dual-axis attention whose two pathways are averaged with equal weight, and dual-axis attention with the learned gate.

Table 3: Resolution study on synthetic families, relative MASE; best single width in bold. Families are defined by measured signal statistics. Average equally weights the three encoders.

Signal family	Patch 2	Patch 4	Patch 8	Average	Multiscale
High-frequency oscillations	**0.888**	0.895	0.976	0.875	0.870
Wide multivariate schemas	**0.405**	0.435	0.427	0.387	0.379
Sparse impulses	0.873	**0.815**	0.970	0.851	0.809
Volatility bursts	0.644	**0.557**	0.776	0.586	0.563
Low signal-to-noise	0.702	**0.473**	0.711	0.602	0.473
Regime-persistent levels	0.531	0.530	**0.486**	0.497	0.484

Dual-axis attention is the strongest operator on all three measures, reducing error against full attention by 9.7% in MASE, 8.4% in CRPS, and 9.1% on multivariate configurations. Restricting interactions to the two grid axes is therefore a useful inductive bias rather than a loss of expressiveness: full attention can represent the same relations but must learn them from data, which it does less effectively at a matched training budget. The fixed-gate variant sits between the two, so the gain splits: restricting attention to the grid axes is worth 4.5% and learning the balance a further 5.5%. The larger share comes from the learned gate, nine scalars that let the two pathways contribute unequally at each depth. Dual-axis attention also cuts peak activation memory by more than half, allowing larger per-device batches (Appendix A1).

7.2 Axis-Aware Two-Dimensional RoPE

We compare four arms under one dual-axis encoder on the same data: no positional encoding, the flattened index $p = tN + n$, temporal-only RoPE with a learned variable bias, and axis-aware two-dimensional RoPE. Removing positional information costs 85% relative MASE, far beyond any difference among the encoded arms, so recovering the grid geometry is the dominant effect. Axis-aware RoPE is strongest, but its margin over temporal-only RoPE is not significant on aggregate GIFT-Eval, which is dominated by univariate and narrowly multivariate series, while on targeted suites it leads by 1.4% on covariate-driven series and by 8.9% on schemas wider than any seen in training, both significant. Explicit variable geometry therefore earns its place where the variable axis carries information (Appendix D1).

7.3 Horizon Scalability

We compare four patch-width-two encoders at matched capacity and update budget, differing only in the horizon objective: fixed blocks of 1024 and of 512 timesteps, lead-time reweighting, which supervises the wide block under a weight decaying with lead time, and horizon sampling from the curriculum of Appendix C1. A fixed block is correct for one span only, so neither fixed arm wins: the wide one never supervises short requests on their own, and halving it improves short horizons by 4.5% but pushes medium and long spans into iterative rollout, losing more than it gains. Reweighting the wide block toward early lead times recovers too little to cover its long-range cost and ends level with both. Horizon sampling is the best of the four and the only arm below both fixed baselines; reweighting the loss toward short horizons is insufficient, and the model must be supervised on them directly (Appendix D2). The released model beats every arm on medium and long horizons and trails on short, the gap Section 7.4 narrows.

7.4 Multiresolution Forecasting

Different patch widths resolve different kinds of structure. Table 3 scores all three on synthetic families grouped by measured signal statistics (Appendix D3), and none wins more than half. Width 2 wins where the informative structure is finer than a wide token, in series that oscillate within it and in wide schemas that leave each variable a short span. Width 4 wins on the event-driven families, wide enough to describe an event yet narrow enough to keep two apart. Width 8 wins on regime-persistent levels, reaching back far enough to see the switches.

Effective combination must therefore be instance-adaptive, which is what the mixing head of Section 4.6 supplies. Each family pools many series, so an allocation that varies within a family can fall

below every fixed width. The last two columns separate adaptation from the mere presence of several resolutions. The learned head beats an equal-weight blend of the same encoders on every family, and matches or beats the best single width on all but one; restricting MORPHEUS-MULTISCALE to a single patch size costs accuracy on both metrics. Two controls at matched capacity of about 247M test whether the gain comes from the added parameters alone or from combining any three models (Appendix D4). Scaling a single patch-4 encoder to 247M is 21.6% worse than the 81.4M standalone, so added parameters alone do not account for the gain. A mixing head over three patch-4 encoders improves on the standalone model, and substituting widths 2, 4, and 8 for those encoders reaches 0.653, 5.2% below the standalone model. Patch sizes coarser than 8 add no information the retained set lacks.

7.5 SYNTHETIC DATA AND TRAINING CURRICULUM

Observed series are hard to learn from directly: periodicity, trend, and cross-variable coupling arrive entangled and buried under measurement noise and missingness, so a transformer trained on real data alone must acquire every pattern at once. A curriculum avoids that but needs simpler material to start from, which is what MySTiC supplies: its underlying factors are specified, so each pattern can be presented in isolation and free of that irregularity, and training can run from purely synthetic to predominantly real. Three parameter-matched arms isolate the two contributions (Appendix D5): adding MySTiC as an undifferentiated mixture improves relative MASE by 6.7%, and ordering the same corpus into four stages yields a further 9.7%, 15.8% in total.

8 LIMITATIONS AND FUTURE WORK

Accuracy is not uniform across the benchmark. Among the systems compared by domain, MORPHEUS-MULTISCALE leads the largest domain, Energy, and Transport, and trails on Econ/Fin and Healthcare; the probabilistic gap concentrates in Healthcare, Web/CloudOps, and Nature and is one of interval width rather than location, with nominal 80% intervals covering 57.7% of observations (Appendix E4). Three properties of the method bound these results: requests beyond the native block are reached by iterative rollout and accumulate error, the fixed token budget of the interleaved canvas trades context depth for the number of variables, and running one encoder per patch width makes MORPHEUS-MULTISCALE costlier at inference than the standalone model.

Each points to a direction. The patch widths are fixed by hand, where the granularity a series needs could instead be learned during training; removing the rollout penalty means supervising the model on context it generated itself; and modeling dependence across targets and lead times would replace the marginal quantile grid with a joint predictive distribution. The mixing head is the most portable of the three: it reads only the context and horizon, never the components' forecasts, and trains from their errors, so a component need only emit aligned quantiles and can otherwise stay frozen or opaque. Alternative backbones (Auer et al., 2025; Ahamed et al., 2026) and published systems (Ansari et al., 2025; Jain & Sen, 2026) all meet that interface, and Appendix E1 shows where the gain would be, since our weakness is concentrated at short horizons.

9 CONCLUSION

MORPHEUS-2.0 treats the three ways forecasting tasks differ, in schema, horizon, and temporal resolution, as modeling problems rather than configuration choices. Dual-axis attention relates variables to one another and each variable to its own history inside every layer, with a learned gate setting their balance at each depth; prediction-length randomization trains one parameter set across horizons spanning two orders of magnitude at no inference cost; and resolving a series at several patch widths under a context-conditioned weighting improves both point and probabilistic accuracy over the standalone model. The two releases give complementary parameter budgets, an 81M-parameter model competitive with systems several times its size and a larger one that attains the lowest relative MASE of any published system on GIFT-Eval, with efficiency claimed as parameter count at matched accuracy rather than latency or memory. All three mechanisms are portable, asking only that a model keep both grid coordinates, sample its training horizon, or emit aligned quantiles, and we offer them as design principles for future time-series foundation models.

USE OF AI TOOLS

Large language model assistants were used for engineering support and manuscript preparation in this project. Their role is described below.

Experiment orchestration. Assistants helped implement driver scripts for training, evaluation, and hyperparameter sweeps, including asynchronous scheduling as accelerators became available. Their role was limited to implementation and bookkeeping; the authors selected the research questions, experimental comparisons, search ranges, and interpretation of the results.

Performance optimization. Assistants supported runtime and memory profiling and suggested implementation changes such as batch sizes, numerical data types, and multiprocessing for evaluation. The authors reviewed these suggestions and verified their effects through executed runs.

Debugging. Assistants were used to investigate data-pipeline and schema errors, distributed-training failures, numerical instabilities, out-of-memory conditions, and evaluation defects. Diagnoses and fixes were checked against the failing runs before they were accepted.

Manuscript polishing. Assistants helped improve the clarity and framing of selected sentences and paragraphs. They also checked the LaTeX sources for formatting problems, unresolved cross-references, unbalanced delimiters, inconsistent terminology, and discrepancies between the main paper and appendix. The authors reviewed and approved all changes to the manuscript.

Related work and citations. Assistants helped search for relevant prior work, locate primary sources, and identify citations for methods and systems discussed in the paper. They also helped prepare and format bibliography entries and check that citation keys resolved correctly. The authors reviewed the cited sources and determined the final related-work positioning.

No reported measurement was produced or altered by an AI assistant; every numerical result in the paper and appendix is the output of an executed run. The authors reviewed AI-assisted code before use and take full responsibility for the content of this work.

REFERENCES

Md Atik Ahamed, Mihir Parmar, Palash Goyal, Chun-Liang Li, Qiang Cheng, Tomas Pfister, and Jinsung Yoon. Reasoning-aware training for time series forecasting. *arXiv preprint arXiv:2605.08625*, 2026. URL https://arxiv.org/abs/2605.08625.

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. URL https://arxiv.org/abs/2410.10393.

Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C. Maddix, Syama Rangapuram, David Salinas, Erik Schultheis, and Lorenzo Stella. GluonTS: Probabilistic time series models in python. *arXiv preprint arXiv:1906.05264*, 2019. URL https://arxiv.org/abs/1906.05264.

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, 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://arxiv.org/abs/2403.07815.

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.

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. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.23719.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza Ramirez, Max Mergenthaler-Canseco, and Artur Dubrawski. N-HiTS: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6989–6997, 2023. doi: 10.1609/aaai.v37i6.25854.

Ben Cohen, Emaad Khwaja, Youssef Doubli, Salahidine Lemaachi, Chris Lettieri, Charles Masson, Hugo Miccinilli, Elise Ramé, et al. This time is different: An observability perspective on time series foundation models. *arXiv preprint arXiv:2505.14766*, 2025. URL https://arxiv.org/abs/2505.14766.

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*, 2024. URL https://arxiv.org/abs/2310.10688.

Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. Flex attention: A programming model for generating optimized attention kernels. *arXiv preprint arXiv:2412.05496*, 2024. URL https://arxiv.org/abs/2412.05496.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1): 1–26, 1979. doi: 10.1214/aos/1176344552.

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*, 2024. URL https://arxiv.org/abs/2401.03955.

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

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. In *Advances in Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. URL https://arxiv.org/abs/2105.06643.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*, 2024. URL https://arxiv.org/abs/2403.13298.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multi-dimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019. URL https://arxiv.org/abs/1912.12180.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30016–30030, 2022. URL https://arxiv.org/abs/2203.15556.

Rob J. Hyndman and Yeasmin Khandakar. Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, 27(3):1–22, 2008. doi: 10.18637/jss.v027.i03.

Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006. doi: 10.1016/j.ijforecast.2006.03.001.

Ayush Jain and Rajat Sen. TimesFM-3: A zero-shot foundation model for multivariate forecasting. Google Research Blog, 2026. URL https://research.google/blog/timesfm-3-a-zero-shot-foundation-model-for-multivariate-forecasting/. Published 31 August 2026.

Emaad Khwaja, Chris Lettieri, Gerald Woo, Eden Belouadah, Marc Cenac, Guillaume Jarry, Enguerrand Paquin, Xunyi Zhao, Viktoriya Zhukova, 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.

Roger Koenker and Gilbert Bassett. Regression quantiles. *Econometrica*, 46(1):33–50, 1978. doi: 10.2307/1913643.

Mario Michael Krell, Matej Kosec, Sergio P. Perez, and Andrew Fitzgibbon. Efficient sequence packing without cross-contamination: Accelerating large language models without impacting performance. *arXiv preprint arXiv:2107.02027*, 2021.

Kenneth C. Lichtendahl, Yael Grushka-Cockayne, and Robert L. Winkler. Is it better to average probabilities or quantiles? *Management Science*, 59(7):1594–1611, 2013. doi: 10.1287/mnsc.1120.1667.

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.

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*, 2024a. URL https://openreview.net/forum?id=JePfAI8fah.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024b. URL https://arxiv.org/abs/2402.02368.

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. URL https://arxiv.org/abs/2502.00816.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

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.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1905.10437.

Prathamesh Patil, Amit Varshney, Manoj Cherukumalli, Harsh Deshpande, Leonard Eun, Dushyant Sahoo, and Naren Chittar. MORPHEUS: A foundation model for multivariate time series forecasting. In *FMSD Workshop at the International Conference on Machine Learning (ICML)*, 2025. URL https://openreview.net/forum?id=3q3dI6hwal.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191, 2020. doi: 10.1016/j.ijforecast.2019.07.001.

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://arxiv.org/abs/2409.16040.

Leslie N. Smith. A disciplined approach to neural network hyper-parameters: Part 1—learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018. URL https://arxiv.org/abs/1803.09820.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063. URL https://arxiv.org/abs/2104.09864.

Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*, 2017. URL https://arxiv.org/abs/1711.11053.

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. URL https://arxiv.org/abs/2402.02592.

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*, 2023. URL https://arxiv.org/abs/2205.13504.

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.

Appendix

A Implementation Details

This section gives the implementation details omitted from Section 4 of the main paper, in the same order: the token layout and how the two attention neighborhoods are executed (Section A1), the positional encoding (Section A2), the configuration of every reported model (Section A3), and the packing and optimization settings (Section A4).

A1 Encoder Architecture

Section 4.3 of the main paper defines the encoder and its two attention neighborhoods. This section specifies four details that description omits.

Token layout. Rows are ordered by variable type, past-only covariates first, then known-future covariates, then targets, before time-major interleaving; packed examples carry distinct sample identifiers that the attention masks additionally require to match; and empty patches and padding are excluded from both neighborhoods.

Rollout beyond the native block. Requests longer than the native block are served by iterative rollout. Each pass rebuilds the forecast window as the native block does: known-future rows are threaded through with their supplied values, while past-only covariate rows receive zero placeholders and are marked unobserved, as future target positions are, so no covariate value is imputed. Quantiles are retained from every pass, but the fed-back median is a point estimate, so uncertainty does not propagate across passes.

Sparse execution. Attention over the two neighborhoods is executed rather than masked. A dense kernel would score every pair and then discard most of them, so we use block-sparse FlexAttention (Dong et al., 2024), which affects execution alone and leaves every quantity the model defines unchanged. Intra-time keys share a patch index and inter-time keys share a variable index, so both neighborhoods are rectangular in the interleaved layout and entire 128×128 tiles of the score matrix are skipped, under a chunked mask so that packed examples of differing schema do not impose one global block pattern. Two qualifications apply to the complexity claim: the saving is realized at tile granularity, so a schema narrower than one tile approaches the dense cost, and $TN^2 + NT^2$ counts admissible pairs and does not bound runtime. Memory constrains the attention-operator comparison, since unrestricted attention materializes all $(TN)^2$ scores and more than doubles peak activation memory; the arms therefore differ in admissible per-device batch, and we match the effective batch size with gradient accumulation.

Fusion gate. The learned gate that fuses the two pathways is one scalar per layer, $g_\ell = \mathrm{sigmoid}(\gamma_\ell) = (1 + e^{-\gamma_\ell})^{-1}$ with $\gamma_\ell \in \mathbb{R}$ unconstrained, shared across all tokens and heads. It is initialized at $\gamma_\ell = 0$, giving both pathways weight 0.5, and optimized with the pathway projections under the forecasting loss alone; there is no auxiliary gate objective.

A2 Positional Encoding

Section 4.4 of the main paper defines the rotary scheme; this section explains what its two constants do and why the operator comparison is fair.

A single scalar position $t + \alpha n$ carries two coordinates at once, so a temporal displacement could in principle be mistaken for a variable one. Neither pathway can make that mistake, because each holds one coordinate fixed: the intra-time pathway compares tokens at the same timestep, where $\Delta t = 0$ leaves only $\alpha \Delta n$, and the inter-time pathway compares tokens in the same row, where $\Delta n = 0$ leaves only Δt. An alias would require a pathway that admits both displacements together, and both pathways exclude such pairs. Unrestricted attention is the one arm that does admit such pairs, and it carries the identical positional package, the same rotary phase and the same learned relative-position bias indexed by a distance that keeps the two coordinates apart, so the baseline receives the same positional information.

The variable-axis scale α and the rotary base θ are the two constants the encoding needs, and both are fixed by grid search on a held-out multivariate synthetic suite. α sets how fast the variable axis turns relative to time. It must be small, so that moving one row costs far less phase than moving one timestep and the two displacements remain distinguishable wherever a pathway sees both; scales below $1/25$ are uniformly worse, and $\alpha = 1/25$ is used at every schema width we report. θ sets the range of displacements the encoding can express. RoPE lays down a geometric ladder of frequencies $\omega_r = \theta^{-2r/d_h}$ across the $d_h/2$ rotary pairs of a head of dimension d_h (Su et al., 2024; Heo et al., 2024): the fastest pair turns once every few positions and separates neighbouring patches, while the slowest has wavelength $2\pi\theta$ and is the only one still able to tell apart positions that are far apart. The base therefore trades local resolution against reach, and both extremes reduce accuracy. Too small, and the slowest phase wraps inside the context window, so a distant position is encoded the same as a near one. Too large, and every pair turns slowly, so adjacent patches receive almost the same phase and local ordering is lost. The selected base, $\theta = 5000$, lies between these regimes for the context lengths we train on.

A3 Model Configurations and Hardware

Section 5.2 of the main paper states that every reported model shares one encoder specification. Table 4 gives it.

Table 4: Shared encoder specification. Patch width and the token budget P are the only fields that vary across the reported models.

Field	Value
Layers	9 dual-axis blocks
Hidden width	640
Attention heads	20
Head dimension	32
Feed-forward width	4096
Rotary base θ	5000
Variable-axis scale α	0.04
Normalization	context-only per row, arcsinh-stabilized
Quantile levels	21
Max patches	P
Max prediction patches	$P/32$

Standalone MORPHEUS-2.0 sets patch width 4 and $P = 8{,}192$, giving 81.4M parameters.

MORPHEUS-MULTISCALE resolves a series at three patch widths, 2, 4, and 8, each an independently trained instance of the same specification. Their token budgets are $P = 16{,}384$, $8{,}192$, and $4{,}096$ patches, chosen so that all three cover the same 32,768 timesteps of interleaved canvas for a univariate series: 31,744 timesteps of context in every case. Their prediction budgets are likewise 512, 256, and 128 patches, which is 1,024 timesteps of native forecast horizon in every case. The three widths therefore differ only in the granularity at which that shared span is resolved; context length and native horizon are identical. Both capacities shrink as $\lfloor P/N \rfloor$ for a schema of N rows. The scale-fusion formulation itself accepts an arbitrary finite set of widths; the reported artifact resolves $\{2, 4, 8\}$.

All reported models are trained on a single eight-GPU A100-40GB node with standard distributed data parallelism.

A4 Sequence Packing and Optimization

This section expands the optimization summary of Section 5.2 of the main paper. Forecasting examples vary enormously in length and schema width, so padding each to a fixed canvas wastes most of the token budget. We therefore borrow sequence packing from language-model pretraining (Raffel et al., 2020; Krell et al., 2021), concatenating several forecasting examples into one interleaved canvas and giving each a distinct sample identifier, which the attention masks then require to match; an example attends within itself and nowhere else, so packing changes throughput without changing any model output.

The packer is greedy and streaming. It draws examples into a staging buffer of 32, keeps at most 16 canvases open at a time, and places each example into an open canvas that still has room. A canvas is released for training once 85% of its token budget is occupied, which costs a little padding but bounds how long any example waits before it is trained on.

Optimization is AdamW (Loshchilov & Hutter, 2019) in bfloat16 with four packed sequences per device and no gradient accumulation, under a one-cycle schedule (Smith, 2018): peak learning rate 10^{-3}, a 10% rising phase, cosine decay to 4×10^{-4}, zero weight decay, and gradient-norm clipping at 10. Because one packed sequence can hold several raw examples, the number of examples per optimizer update varies, so we compare training budgets using realized patch-token exposure per trainable parameter rather than optimizer-step count (Hoffmann et al., 2022).

B Dataset Details

B1 Pretraining Corpora

Section 5.1 of the main paper introduces the real half of the pretraining mixture; this section measures it. The real portion of the mixture is GIFT-Eval Pretrain, the public pretraining corpus released with the benchmark (Aksu et al., 2024), which excludes its test windows, so training on it does not leak evaluation targets. Table 5 reports its composition: 152 dataset partitions across the seven GIFT-Eval domains and 13 sampling frequencies, 28.4 million series, 229 billion observations, and 938 GB on disk. A series here is one target row of one item, so a partition of 307 items with 3 targets contributes 921 series.

Table 5: Composition of GIFT-Eval Pretrain. Series counts target rows, not items; size is on-disk Arrow. Domain assignment follows the GIFT-Eval taxonomy, and every partition is assigned to exactly one domain.

Domain	Partitions	Series	Obs. (B)	Size (GB)	Frequencies
Nature	70	25,397,091	204.5	818.2	H, 6H, D
Energy	29	1,804,428	16.9	68.9	4S, 1min, 10min, 30min, H
Transport	24	165,266	4.9	19.6	5min, 15min, 30min, H, D, W
Web/CloudOps	6	900,863	2.6	30.5	5min, D, W
Sales	4	148,654	0.2	0.8	D, M
Healthcare	3	1,929	< 0.1	< 0.1	W
Econ/Fin	16	5,555	< 0.1	< 0.1	D, W, M, Q, Y
Total	**152**	**28,423,786**	**229.1**	**938.0**	13 distinct

The corpus is severely imbalanced, which motivates a sampling policy distinct from its volume distribution. Climate reanalysis alone, the ERA5 and CMIP6 partitions within Nature, accounts for 87% of bytes and 89% of series, whereas Healthcare and Econ/Fin together contribute under 0.1% of either. Volume-proportional sampling would therefore concentrate capacity on a single domain and under-represent the low-frequency monthly, quarterly, and yearly regimes that only the smallest partitions supply. We instead assign each domain–frequency combination an explicit sampling weight, set from the size and composition of that combination, not in proportion to it: the largest sources are held down so that no single one can dominate a batch, and the smallest are raised enough to appear regularly without being repeated so often that the model simply memorizes them. Every domain–frequency combination therefore retains non-negligible mass in each batch. The synthetic portion described in Section B2 is added to this mixture.

B2 Synthetic Data Generation

Section 5.1 of the main paper introduces MySTiC in three sentences; this section gives the generators behind it. Public forecasting corpora are large but narrow, having grown largely by restacking a common pool of sources (Godahewa et al., 2021). As Section B1 shows, volume concentrates in a few sources and a few frequencies, so most of the space a universal forecaster has to cover is thinly populated: wide panels occur at only a handful of frequencies, covariate relations are almost never labelled, and nonstationarity arrives in whatever degree a source happens to contain. MySTiC is a

generated corpus built to widen that coverage and to make the curriculum of Section B3 possible. Its factors are specified, not inferred, so each kind of temporal structure can be presented on its own, free of the measurement noise and missingness that accompany it in observed data. That gives the early curriculum stages material a model can learn from before it has to contend with real series, which makes training from random initialization tractable.

Three subsets divide the work, one per subsection below.

Every family is required to be forecastable: each carries a repeating waveform, a persistent level, or a recoverable covariate relation, and noise is added on top of that structure, not substituted for it. This excludes unpredictable series, on which no model can improve over a trivial baseline. Records use the model's variable–time layout with variable types assigned during generation, and deterministic seed keys make individual records reproducible. Table 6 reports the measured training copy.

Table 6: MySTiC composition. A record is one multivariate series and rows are its variables.

Subset	Collections	Records	Rows/record	Obs. (B)	Size (GB)
Kernel composition (§B2.1)	42	357,000	1–21	5.7	24.1
Covariate response (§B2.2)	14	365,004	1–5 targets	2.3	34.9
Real-world emulation (§B2.3)	82	585,890	1–21	14.3	61.6
Total	**138**	**1,307,894**		**22.3**	**120.6**

B2.1 KERNEL-COMPOSITION SUBSET

This subset exists to vary periodicity, trend, memory, and cross-row dependence independently of one another, extending the generator of Morpheus (Patil et al., 2025). In observed panels those properties arrive bundled, since a given source fixes its width, its sampling frequency, and its correlation structure together. Here each can be set separately, so the subset supplies structural breadth across domains and suits the first curriculum stage, where the model should meet each temporal primitive on its own.

The univariate backbone is a Gaussian-process prior (Rasmussen & Williams, 2006) with a sampled covariance function, composed from base kernels for periodic components, trends, decaying correlations, and noise, and evaluated on a reference-scaled grid so that a configured cycle keeps its intended period across series lengths. A row combines its kernels under sampled sum or product operators,

$$\kappa \;=\; \big(\cdots(\kappa_1 \diamond_1 \kappa_2)\diamond_2 \cdots\big) \diamond_{J-1} \kappa_J, \qquad \diamond_j \sim \mathcal{U}\{+,\times\}, \tag{5}$$

and is drawn from the resulting prior. Summation superposes components while multiplication makes periodic amplitude vary with a trend, so the same components express both seasonality on a baseline and locally periodic behaviour whose amplitude drifts. Long series instead specify a power spectrum directly, which varies how energy is distributed across frequencies independently of the covariance composition.

Cross-row dependence enters two ways: latent spectral factors g_f give each row a shared and an idiosyncratic component, and a lead–lag construction makes one row follow a driver d after delay k,

$$y_i \;=\; w_i\, x_i \;+\; \sum_f \lambda_{if}\, g_f, \qquad w_i + \textstyle\sum_f \lambda_{if} = 1, \tag{6}$$

$$y_i(t) \;=\; c\, x_d(t-k) \;+\; (1-c)\, x_i(t) \;+\; \epsilon_i(t). \tag{7}$$

The first produces a low-rank common component and the second a delayed propagation between rows; the two can be combined in one panel. Level shifts applied to all rows or to a subset, a slow variance envelope, and per-row rescaling then add nonstationarity and scale variation, so predictive intervals must adapt over time instead of holding constant width. Figure 2 shows six records drawn from this subset.

B2.2 COVARIATE-RESPONSE SUBSET

Public corpora rarely label covariate relations: a real dataset supplies covariate columns without identifying which ones matter, at what lag, or how strongly. This subset fixes that relation by con-

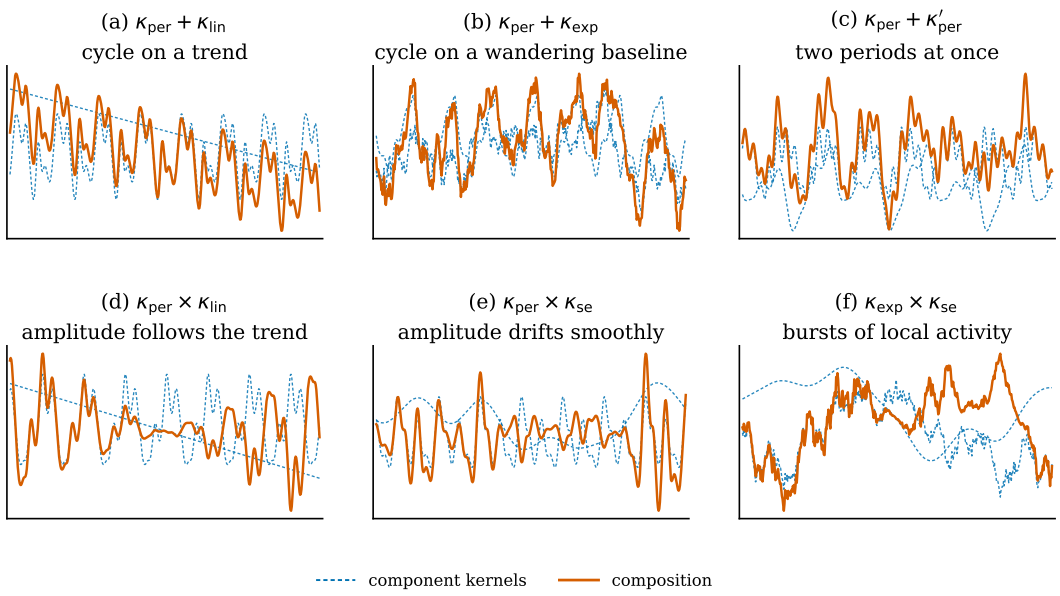


Figure 2: Kernel composition in the MySTiC kernel subset. Each row is drawn from a Gaussian process whose covariance is assembled from base kernels under the sums and products of Eq. 5: a sum superposes independent structure, and a product modulates one component by another. Panels combine periodic, linear, exponential, and squared-exponential kernels (κ_{per}, κ_{lin}, κ_{exp}, κ_{se}), with the component draws in blue and the composition they produce in red. Sums give a cycle on a trend (a), a cycle on a wandering baseline (b), and two periods at once (c); products give periodic amplitude that follows a trend (d) or drifts smoothly (e), and bursts of local activity (f). Series are standardized for display.

struction, so that covariate-conditioned supervision is available and prediction quality can be compared against an oracle receiving the same visible inputs, which separates failures of covariate use from irreducible noise. Covariate rows are standardized draws from periodic, piecewise-constant, or autoregressive processes, so informativeness lies in the response relation, not the input scale.

Each target is then assigned one response recipe and a driving subset of the covariate rows $x_1, \ldots, x_k$:

$$\text{instantaneous linear:} \quad y_t = \sum_i w_i\, x_i(t) + \varepsilon_t, \tag{8}$$

$$\text{instantaneous nonlinear:} \quad y_t = \sum_i w_i\, x_i(t) + \sum_j g_j\big(x_{i_j}(t)\big) + \sum_{(i,j)} w_{ij}\, x_i(t) x_j(t) + \varepsilon_t, \tag{9}$$

$$\text{lagged linear:} \quad y_t = \sum_i w_i\, x_i(t - k_i) + \varepsilon_t, \tag{10}$$

$$\text{distributed lag:} \quad y_t = \sum_i \sum_{s=k_i^{\text{lo}}}^{k_i^{\text{hi}}} a_i\, \rho_i^{\,s-k_i^{\text{lo}}}\, x_i(t-s) + \varepsilon_t, \tag{11}$$

$$\text{ARX:} \quad y_t = \phi\, y_{t-1} + \sum_i \beta_i\, x_i(t) + \varepsilon_t, \tag{12}$$

The g_j are scalar nonlinearities from a fixed bank of saturating, rectifying, and periodic maps, and the $w_{ij} x_i x_j$ terms are multiplicative interactions, which make one driver's effect depend on the level of another and so cannot be represented by any additive function of the covariates. The recipes differ in what a forecaster must do to exploit them: the instantaneous forms require reading across rows at a single time index, the lagged forms add temporal alignment, the distributed variant spreads one input's effect over a decaying window, and the autoregressive model with exogenous inputs (ARX)

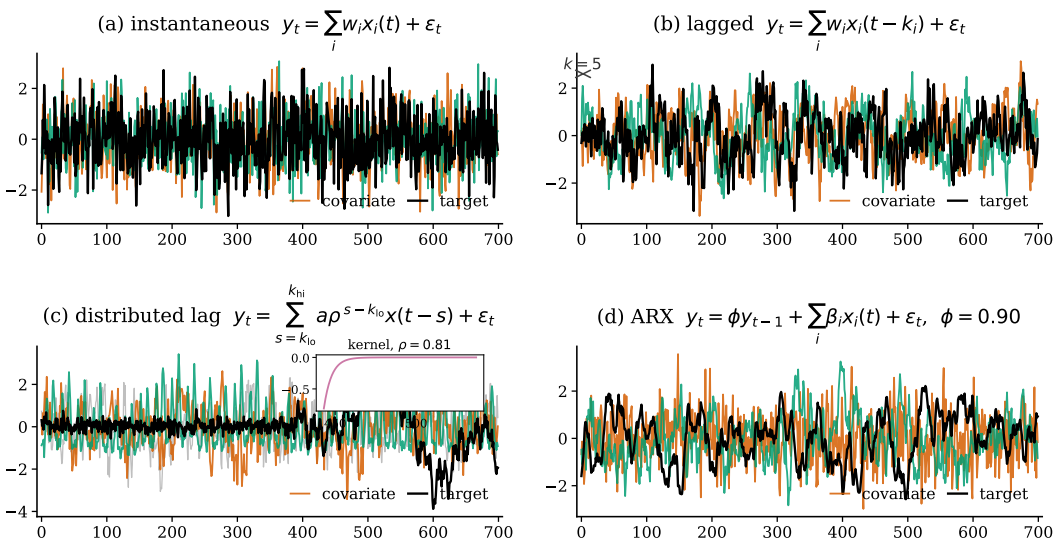


Figure 3: Covariate-response subset. Black is the target, colored lines are the covariates that drive it, and grey lines are distractors that drive nothing; all rows are standardized for display. (a) instantaneous linear response, Eq. 8; (b) lagged response, Eq. 10, with the realized lag k annotated; (c) distributed lag, Eq. 11, with the geometric kernel inset; (d) ARX, Eq. 12, with its persistence parameter ϕ annotated.

makes the target depend on its own previous value as well as the current covariates. Figure 3 shows one record for each of four recipes, with the driving covariates and the inert distractors distinguished by colour.

Response coefficients have a nonzero magnitude floor and noise ε_t is scaled relative to the realized deterministic signal, so signal-to-noise stays comparable across recipes. Known-future rows remain visible over the horizon while past-only rows stop at the forecast origin, so a past-only driver is usable only when its lag places the needed values in context, and both usable and unavailable cases are generated. Distractor rows drawn from the same marginal processes drive no target, so marginal appearance cannot identify a relevant input. Each record retains its response specification, driver indices, and lags for oracle scoring.

B2.3 Real-World Emulation Subset

This subset adds a repeating conditional mean, which the first two subsets do not control directly. A fixed within-period shape is modulated across periods and combined with residual variation, which differs from spectrum-matched smooth noise in retaining a mean that returns at a characteristic period, the structure seasonal baselines rely on.

Periods are specified in wall-clock time and converted to samples using each series' sampling frequency, so one profile can be rendered at any frequency while preserving its phase, and one shape appears at many ratios of cycle length to patch width. Two primitives build the waveform on the circular within-period phase, a trapezoid that rises on a cosine ramp, holds flat, and falls, and a circular bump; between them they express pulses with exact zeros, flat-topped plateaus, and smooth humps, shapes a single sinusoid does not represent. Each period d is then scaled by

$$\alpha_d \;=\; a_d \cdot \omega_{d \bmod 7} \cdot e_d, \tag{13}$$

where a_d is persistent amplitude variation, ω modulates a weekly cycle, and e_d is a piecewise-constant activity state, so period-to-period dependence and abrupt transitions both survive. Optional components add multiplicative texture, level variation, trend, residual noise, and phase-consistent spikes, applied so that structural zeros and forecastable timing survive. Where a record contains a break, it is placed so that forecast windows can straddle it, and in multivariate records breaks share timing while differing in per-row magnitude. Figure 4 shows six records drawn from this subset.

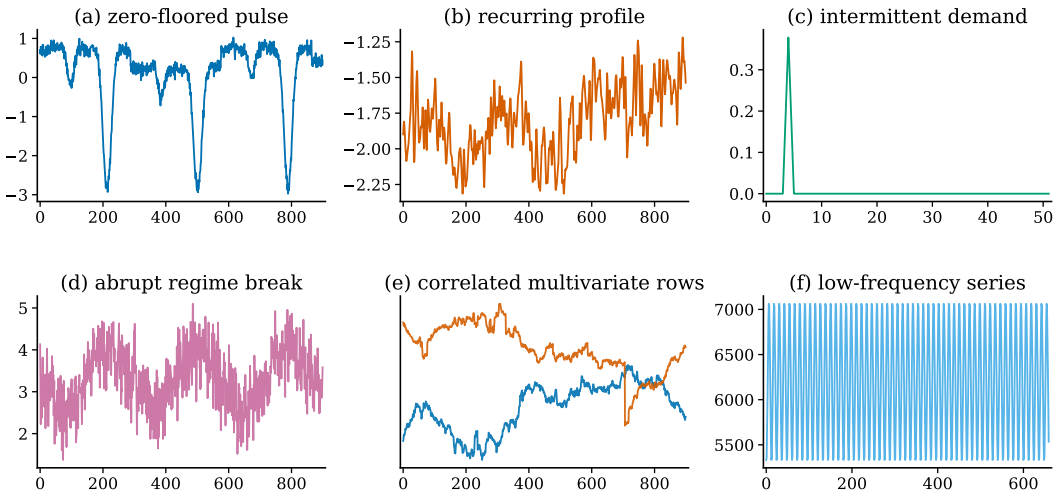


Figure 4: Real-world emulation subset. (a) zero-floored pulse from the trapezoid primitive, with exact zeros between cycles; (b) a recurring profile under the amplitude modulation of Eq. 13; (c) a short intermittent record; (d) an abrupt regime break; (e) a correlated multivariate record, rows offset vertically for legibility; (f) a low-frequency series. Panels are individual records.

B3 Data Mixture and Curriculum Schedule

This section gives the stage shares behind the curriculum of Section 5.2 of the main paper. The two corpora above are not presented as a single undifferentiated pool. The mixture is divided into four stages that increase distributional complexity and realism monotonically, and the share of real observations rises across them while the synthetic share falls. The first stage takes 15% of the update budget and is dominated by the kernel-composition subset of Section B2, whose periodicity, trend, and cross-row dependence are specified, not inferred, and therefore appear in isolation and free of measurement noise. The second takes 20% and mixes all three synthetic subsets, remaining synthetic-dominated while admitting a small amount of GIFT-Eval Pretrain. The third moves to roughly equal synthetic and real proportions, and the fourth is approximately 90% real, adapting the model to the missingness, irregularity, and heterogeneity of observed series. Within the real portion every stage retains the domain–frequency sampling weights of Section B1, so the curriculum reweights the synthetic and real blocks against each other without changing the balance inside either.

Section D5 reports what each ingredient contributes.

C Training and Evaluation Protocols

C1 Prediction-Length Curriculum

This section gives the sampling distribution behind Section 4.5 of the main paper. Every reported patch width is trained with prediction-length randomization. The non-uniform curriculum samples

$$\{m_k\} = \{4, 6, 15, 24, 240, 360, 512\}, \qquad \{\pi_k\} = \{.04, .03, .06, .12, .20, .20, .35\}. \tag{14}$$

One M is shared by every example in a batch, while the timestep horizon remains schema dependent through $H_i(M) = s \max(1, \lfloor M/N_i \rfloor)$. This preserves compatible packed layouts. The curriculum varies masked ground-truth horizons; it does not expose the model to its own recursively generated context and should not be interpreted as eliminating autoregressive exposure bias.

C2 Mixing-Head Training

Section 4.6 of the main paper defines the mixing head and states that it is trained after the encoders; this section gives its architecture and objective. The encoders at each patch width are trained first; the mixing head, the context-conditioned encoder that combines them, is trained in a final stage on

top of them on a real subset of the pretraining corpus. Encoder parameters receive no gradient from this stage, so the resolutions entering the combination are exactly the models scored individually elsewhere in this appendix. Figure 5 sketches the result.

The head maps the forecast-origin context and the requested horizon to a distribution over the E components. It segments the context into patches of 8 steps, up to 4096 patches spanning 32,768 timesteps, and embeds the interleaved variable–time grid of target, past-only covariate, and known-future covariate rows with a finite-value mask. One dual-axis block of width 128, 4 heads, and a 512-unit feed-forward sublayer then encodes the tokens, with rotary encoding on the intra-series axis. A readout pools them into a fixed-width summary from four learned horizon-conditioned attention queries and their masked mean, maximum, and second moment; a horizon embedding is injected into the queries and re-concatenated before a two-layer head emits coefficients $p_e = \mathrm{softmax}(z)_e$. Pooling is what makes the head schema- and length-agnostic, since queries and masked moments both reduce a variable token count to fixed width. The head totals 1.2M parameters, under 0.5% of the inference budget.

Supervision comes only from errors the components actually realize. For window i, let $\mathrm{err}_{i,e}$ be the mean absolute scaled error of component e, let $\mathrm{err}_i^\star = \min_e \mathrm{err}_{i,e}$, and let $\delta_{i,e} = (\mathrm{err}_{i,e} - \mathrm{err}_i^\star)/\mathrm{err}_i^\star$ be the relative regret, so a window on which the components score 1.00 and 1.02 counts the same as one on which they score 50 and 51. Keeping the signal scale-free prevents a few large-magnitude windows from dominating the objective. A temperature λ maps regrets to a detached target on the simplex, against which the head is trained by cross-entropy:

$$q_{i,e} = \frac{\exp(-\delta_{i,e}/\lambda)}{\sum_{e'} \exp(-\delta_{i,e'}/\lambda)}, \qquad \mathcal{L}_{\mathrm{soft}} = -\frac{1}{|\mathcal{W}|} \sum_{i \in \mathcal{W}} \sum_{e} q_{i,e} \log p_{i,e}, \qquad \frac{\partial \mathcal{L}_{\mathrm{soft}}}{\partial z_i} = p_i - q_i. \quad (15)$$

Because q is detached, this differs from $\mathrm{KL}(q \,\|\, p)$ only by the constant $H(q)$ and has an identical gradient. Optimization is AdamW with warm-up and cosine decay, bf16 autocast, and multi-GPU data parallelism, with batches assembled under a fixed value budget so that long and short contexts pack efficiently; λ is held small.

Three properties motivate this construction. It is *label-free*: the target comes from the error vector and never forms an $\arg\min$, so near-ties are not forced into an arbitrary class, precisely where a hard classification surrogate injects label noise. It is *strictly proper and interior* (Gneiting & Raftery, 2007): cross-entropy is logarithmic in p, so the unique minimizer is $p_i = q_i$, whereas an objective linear in p is minimized at a vertex of the simplex and admits a degenerate, context-independent solution placing all mass on the globally strongest component. The soft objective has no such degenerate minimizer, so the head's weights remain context-dependent. It also *adapts its confidence*: near-ties give a diffuse target that teaches the head to hedge, large gaps a near one-hot target that teaches it to commit. The temperature spans the family's two limits, $\lambda \to 0$ collapsing the target onto the $\arg\min$ label and $\lambda \to \infty$ flattening it to no preference.

We measure this on held-out windows: the soft target improves on hard-label cross-entropy by 2.4% relative MASE and 3.9% expected regret, and combining all E components beats keeping the top two, which beats hard top-one selection.

C3 Metric Definitions and Aggregation

This section defines precisely the metrics named in Section 5.3 of the main paper. The primary head uses the 21-level grid

$$\mathcal{Q} = \{.01, .05, .10, .15, \ldots, .90, .95, .99\}. \quad (16)$$

For target j and lead time h, the model estimates the conditional quantile function

$$Q_{j,h}(\tau \mid \mathcal{C}_T) = \inf\{q : \Pr(y_{j,T+h} \le q \mid \mathcal{C}_T) \ge \tau\}, \quad (17)$$

the one-dimensional marginal of the conditional law of $\mathbf{y}_{T+1:T+H}$ given $\mathcal{C}_T$. The implementation multiplies the conventional pinball loss (Koenker & Bassett, 1978) by two, a constant that does not change its minimizer. It does not impose a non-crossing constraint, and its outputs are marginal quantiles for each target and lead time rather than a joint multivariate distribution. Evaluation linearly interpolates native levels onto the benchmark grid when necessary. Requests longer than the native block are completed by iterative rollout, feeding back the predicted medians and retaining the quantiles of each pass.

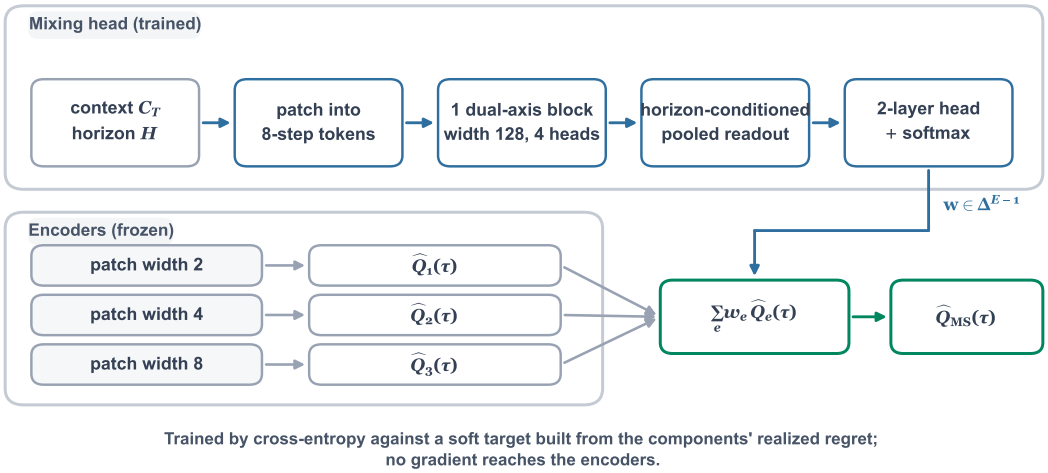


Figure 5: Architecture of the mixing head. The forecast-origin context is patched, embedded with its observation mask, and encoded by one dual-axis block; a horizon-conditioned pooled readout reduces the tokens to a fixed-width summary from which a two-layer head emits one weight per patch width. The encoders are frozen at this stage and the head never reads their forecasts, so the weights depend on the request alone.

Benchmark scoring uses the 9-level grid $\mathcal{Q}_G = \{0.1, 0.2, \ldots, 0.9\}$ (Aksu et al., 2024). For configuration c, let Ω_c contain all valid item–target–horizon values, and let i index an item–target pair, the benchmark's counterpart to the target index j of the main paper's problem formulation, with history length T_i, seasonal period m_i, and historical seasonal scale sn_i. The two headline metrics are

$$\mathrm{MASE}_c = \frac{1}{|\Omega_c|} \sum_{(i,h)\in\Omega_c} \frac{|y_{i,h} - \widehat{Q}_{i,h}(0.5)|}{\mathrm{sn}_i}, \qquad \mathrm{sn}_i = \frac{1}{T_i - m_i} \sum_{t=m_i+1}^{T_i} |y_{i,t} - y_{i,t-m_i}|, \tag{18}$$

$$\mathrm{MWQL}_c = \frac{1}{9} \sum_{\tau\in\mathcal{Q}_G} \frac{2\sum_{(i,h)\in\Omega_c} \rho_\tau(y_{i,h} - \widehat{Q}_{i,h}(\tau))}{\sum_{(i,h)\in\Omega_c} |y_{i,h}|}, \quad \rho_\tau(u) = \max\{\tau u, (\tau-1)u\}. \tag{19}$$

The continuous ranked probability score (CRPS) rewards a predictive distribution for concentrating near the observed value. The GIFT-Eval quantity called CRPS is operationally its normalized aggregate of mean weighted quantile loss, rather than a numerical integral of a continuous predictive CDF. Headline metrics use all evaluation items and geometric means over the 97 configurations. Writing L_c for a metric on configuration c, the reported relative value is

$$\mathrm{Rel}(L) = \frac{\exp\left(\frac{1}{97}\sum_c \log L_c\right)}{\exp\left(\frac{1}{97}\sum_c \log L_c^{\mathrm{SeasonalNaive}}\right)}, \tag{20}$$

so Seasonal Naive scores 1.0 by construction. The reference predictor repeats the observation from one native-frequency seasonal period earlier. Its unnormalized geometric means over the 97 configurations are 1.398 for MASE and 0.252 for MWQL, which recover absolute values from any relative figure reported in the paper. Within each configuration, GluonTS (Alexandrov et al., 2019) aggregates MASE over all valid item, target, and lead-time values; MWQL first sums quantile losses and absolute labels over the same values and then forms their ratio. Seasonal periods are inferred from the native frequency. If the nominal period exceeds the available history, GluonTS uses a lag-one scale for MASE (Hyndman & Koehler, 2006).

C4 Evaluation Setup Categories

This section defines the Setup column used in the result tables of Section 6 of the main paper. Submissions are distinguished by which GIFT-Eval data a model was exposed to before scoring, and we use three labels throughout. Zero-shot models are trained without any GIFT-Eval data,

including its training split. Pretrained models may include the GIFT-Eval training split in their pretraining mixture and are then scored with one set of weights across all configurations. Fine-tuned models additionally adapt a pretrained checkpoint to the benchmark training split. All three categories exclude the GIFT-Eval test windows; the distinction concerns the training split only.

MORPHEUS-2.0 and MORPHEUS-MULTISCALE are trained on GIFT-Eval Pretrain, which includes the GIFT-Eval training split, and are scored with one set of weights across all 97 configurations without per-configuration adaptation. They are therefore reported as pretrained on the benchmark. Table 13 records the category of every system it lists, taken from that system's own public reporting.

D ADDITIONAL ABLATIONS

D1 POSITIONAL ENCODING ABLATIONS

This section gives the arms and measurements summarized in Section 7.2 of the main paper. Table 7 places the arms structurally before Table 8 scores them. The two named alternatives are temporal RoPE with learned variable identifiers (Su et al., 2024) and alternating axial attention (Ho et al., 2019; Zhang & Yan, 2023).

Table 7: Structural comparison of positional and attention designs; entries summarize inductive biases.

Design	Time geometry	Variable geometry	Both axes per layer
Flattened one-dimensional positions	Entangled with N	Entangled with N	Model dependent
Temporal RoPE + variable identifiers	Relative	Absolute	Model dependent
Alternating axial attention	Relative	Relative	No
Ours	**Relative**	**Relative**	**Yes**

Table 8: Positional-encoding error on aggregate, covariate-informed, and held-out-width suites. Lower is better.

	All configurations		Covariate-informed	Held-out width
Position scheme	Rel. MASE	Rel. CRPS	Rel. MASE	Rel. MASE
Axis-aware two-dimensional RoPE	**0.689**	**0.522**	**0.708**	**0.713**
Temporal RoPE + variable bias	0.699	0.534	0.718	0.783
Flattened one-dimensional RoPE	0.704	0.543	0.741	0.732
No positional encoding	1.276	0.932	1.118	2.544

All four arms train on the same data under the same recipe. Only the temporal arm adds parameters, an embedding over variable identifiers worth about 0.04% of the total, since a rotary phase is computed and not learned. No arm's result can be attributed to extra capacity.

Each targeted suite tests one capability. The covariate-informed suite is 7000 synthetic series whose target is a known function of its covariates, drawn from instantaneous, bilinear, gated, past-lag, future-lag, and interaction recipes at widths of at most 8 rows, with generator seeds disjoint from the training roster; scoring well on it requires reading across the variable axis. The held-out-width suite instantiates one synthetic family at $N \in \{2, 4, 6, 8, 12, 16, 21\}$ with 300 series per width, split into an in-range block at $N \leq 8$ and an out-of-range block at $N \in \{12, 16, 21\}$ that no arm saw in training. Reported values are geometric means over each suite. Against the temporal-plus-bias arm, the axis-aware scheme leads by 1.4% (95% paired bootstrap CI $[1.0, 1.8]$) on the covariate-informed suite and by 8.9% (95% CI $[7.5, 10.5]$) on the held-out-width suite, whereas its 1.4% aggregate margin on GIFT-Eval has an interval spanning zero.

D2 HORIZON OBJECTIVES

Table 9 gives the per-group breakdown of the horizon objectives compared in Section 7.3 of the main paper. All four arms are patch-width-two encoders at matched capacity and update budget,

with every other training setting held fixed, differing only in the objective: two fixed baselines supervising a constant block, lead-time reweighting supervising the wide block under a weight that decays with lead time, and horizon sampling drawing the prediction-token budget from the curriculum of Section C1.

Table 9: Horizon objectives at matched capacity. Entries are relative MASE within each horizon group; lower is better.

Horizon objective	Supervised span	Short	Medium	Long	Overall
Fixed prediction length	1024	0.734	0.673	0.659	0.704
Fixed prediction length	512	0.701	0.714	0.716	0.707
Lead-time loss reweighting	1024 (weighted)	0.721	0.689	0.683	0.706
Horizon sampling (MORPHEUS-2.0)	8–1024	0.706	0.687	0.678	0.696

D3 PATCH RESOLUTION

This section defines the signal families scored in Section 7.4 of the main paper. Families group collections by signal statistics measured on the target series before any forecast is scored, and every collection appears in exactly one family. Table 10 gives the defining statistic of each.

Table 10: Signal-family definitions for the resolution study. Every statistic is measured on the target series before any forecast is scored.

Signal family	**Defining statistic**
High-frequency oscillations	Ten-second and five-minute sources; a third of spectral energy lies above the band a width-8 patch can represent
Wide multivariate schemas	7 to 21 simultaneous targets
Sparse impulses	Differenced kurtosis above 300 at a jump rate below 0.04: isolated spikes on a smooth baseline
Volatility bursts	Absolute-difference autocorrelation near 0.3 with a jump rate above 0.12
Low signal-to-noise	Lowest lag-1 autocorrelation and flattest spectra in the corpus
Regime-persistent levels	Lag-50 autocorrelation between 0.69 and 0.82

All three patch sizes are scored together from one code path on identical windows in identical order, so per-series values are paired, and each cell is the geometric mean of that patch size's MASE over the family, relative to Seasonal Naive.

D4 MIXING-HEAD CONTROLS AND SINGLE-WIDTH RESTRICTIONS

This section reports in full the two capacity-matched controls cited in Section 7.4 of the main paper, together with the single-width restrictions. Two controls separate the effect of multiple granularities from two alternative explanations: a single encoder at patch width 4, made both wider and deeper so that its parameter count matches the full MORPHEUS-MULTISCALE budget, which tests whether the gain is bought with parameters, and a mixing head over three independently trained encoders that share one patch size, which tests whether it is generic ensembling. Table 11 reports both at a matched inference budget of about 247M, alongside the three encoders scored on their own. The three single-width rows differ slightly in parameter count only because the learned temporal relative-position bias is indexed by patch distance, so a narrower patch carries a larger bias table.

No single width is best on both metrics. Width four attains the lowest relative MASE and width two the lowest relative CRPS, the expected consequence of the granularity trade-off: wider patches reach further into history and help the median forecast at longer lead times, while narrower patches retain the local variation that the quantile spread must capture. The width set was fixed on the synthetic suite of Section D3, where widths coarser than eight added nothing the retained set did not already

resolve. Adding width sixteen under the mixing head does not improve forecasts on any domain or frequency, so the reported model stops at width eight.

Added capacity alone does not explain the gain: at 247M a single patch-4 encoder reaches 0.838 relative MASE, 21.6% worse than the same architecture at 81.4M and worse than every single-width row in the table. Three patch-4 encoders trained from different initializations on the same mixture, combined under the mixing head, reach 0.669 and 0.492, improving on the released patch-4 standalone by 2.9% in MASE and 5.7% in CRPS; because every component resolves the series at the same granularity, that margin is what generic ensembling contributes on its own. Seeds with similar aggregate scores (Section 6 of the main paper) still make different errors on individual series, and by Jensen's inequality (Section 4.6 of the main paper) a weighted average of their quantiles scores no worse than the weighted average of their scores; in our measurement the combination also improves on the released patch-4 encoder. Replacing those three with widths 2, 4, and 8 improves a further 2.4% in MASE and 3.3% in CRPS, to the released 0.653 and 0.476.

The 5.2% MASE gain over the standalone model thus has two roughly equal parts: 2.9% from combining three encoders of the same patch width, and a further 2.4% from giving those encoders different widths. Neither part alone reaches the released 0.653.

Table 11: Isolating the mixing head. Every row uses the same 97 configurations, evaluation protocol, rollout policy, and benchmark quantile grid, so rows differ only in what is combined and at what capacity. Lower is better.

System	Params	Rel. MASE	Rel. CRPS
MORPHEUS-2.0, patch 2	84.3M	0.696	0.507
MORPHEUS-2.0, patch 4 (released standalone)	81.4M	0.689	0.522
MORPHEUS-2.0, patch 8	80.1M	0.708	0.526
Single encoder at multiscale budget, patch 4	247M	0.838	0.602
Mixing head over three encoders, all patch 4	245M	0.669	0.492
Mixing head over patch widths 2, 4, 8 (MORPHEUS-MULTISCALE)	247M	**0.653**	**0.476**

D5 SYNTHETIC DATA AND CURRICULUM

This section gives the three arms behind Section 7.5 of the main paper. The pretraining recipe makes two separable claims: that generated series cover regimes the real corpus underrepresents, and that the order of exposure matters independently of the content. Table 12 isolates each by adding one ingredient at a time. The three arms are parameter matched and share every other training configuration, including optimizer, learning-rate schedule, and update budget; they differ only in the pretraining mixture and the order in which it is presented. The first arm trains on GIFT-Eval Pretrain alone, the second adds MySTiC as a single undifferentiated mixture, and the third presents that same corpus under the four-stage schedule of Section B3.

Table 12: Contribution of the synthetic corpus and the curriculum that orders it. GM MASE is the geometric mean (GM) of MASE over the 97 configurations, and Relative MASE divides it by that of Seasonal Naive. Lower is better; the last row is the released model.

Pretraining recipe	GM MASE	Relative MASE
Real corpus only	1.144	0.818
+ MySTiC, single-stage mixture	1.067	0.763
+ **four-stage curriculum**	**0.963**	**0.689**

The second effect is the larger, and it introduces no new data: rows two and three share generators and differ only in scheduling. The ingredients are also not separately sufficient, since absent generated series a curriculum could only reweight real sources, which is the sampling policy the first arm already applies.

E Additional Results

Every Morpheus-2.0 and Morpheus-Multiscale entry in this section is a mean over five independent training seeds, as in the main paper; all other entries are the values their authors report. The seed standard deviations are given in Section 6 of the main paper.

E1 Extended Comparison and Horizon-Wise Accuracy

Table 13 extends the comparison of Section 6 of the main paper to a wider set of model families and decomposes each one by horizon group. Raw per-group errors are not comparable across groups, so every entry is normalized by the Seasonal Naive geometric mean computed within its own group. Unnormalized geometric means are omitted because they differ from the relative values only by those fixed denominators.

The decomposition separates profiles that the aggregate hides, and the models divide into two families. Chronos-2 (Ansari et al., 2025) is strongest at short horizons and degrades steadily as the horizon grows, ending 13.5% worse at long than at short; Toto-2.0-2.5B, TimesFM-2.5, and Moirai-2.0-R-Small (Khwaja et al., 2026; Das et al., 2024; Woo et al., 2024) follow the same shape. Morpheus-2.0 runs the other way, improving from short to long, so it and Chronos-2 are nearly tied in aggregate while disagreeing at every individual horizon: Chronos-2 leads by 11% on short, and Morpheus-2.0 leads by 20% on long. We attribute this pattern to patch width. A width-4 token resolves less local detail than Chronos-2 retains, which costs short horizons, but reaches further into history per token, which pays at long ones. Falcon-X is the only other entry with that orientation.

Morpheus-Multiscale improves on Morpheus-2.0 in every group and on both metrics, by the widest margin at short horizons where the narrow widths contribute most. It records the best relative MASE in the table overall and the best relative MASE and relative CRPS at both medium and long horizons, where its lead is largest: at long horizons it improves on TimesFM-3 by 14.6% in MASE and 10.5% in CRPS, and the runner-up at that horizon on both metrics is our standalone model Morpheus-2.0.

The spread across horizon groups follows from patch width. A token spanning two or more timesteps reaches further into history but resolves less local detail, and short requests depend on that detail. The mixing head recovers part of the loss, improving on Morpheus-2.0 by 7.8% relative MASE at short horizons, its largest gain in any group, because the narrow widths receive most of the weight there. A deficit against TimesFM-3 (Jain & Sen, 2026), contemporaneous work and the nearest system to ours in aggregate, remains at short horizons; our margins at medium and long offset it, leaving a net 2.1%. Short requests are 55 of the 97 configurations and carry 57% of the weight in the geometric mean, so the aggregate margin holds under a weighting that favours the group where we are weaker.

E2 Per-Configuration Results Across Metrics

Section 5.3 of the main paper reports two headline metrics and defers the rest here. Besides MASE and mean weighted quantile loss, the evaluator records mean and median MSE, median MAE, MAPE, sMAPE, RMSE, NRMSE, and ND for every configuration. We report four of these. The other four add nothing after normalization by Seasonal Naive: relative ND equals relative MAE, relative NRMSE equals relative RMSE, relative RMSE is the square root of relative MSE, and median MSE equals mean MSE because our median serves as the mean forecast. Writing y for an observed future value and $\widehat{y}$ for the median forecast, and taking MASE as defined in Section C3, the four we keep are

$$\mathrm{ND} = \frac{\sum |y - \widehat{y}|}{\sum |y|}, \quad \mathrm{MSE} = \overline{(y - \widehat{y})^2}, \quad \mathrm{MAPE} = \frac{\overline{|y - \widehat{y}|}}{|y|}, \quad \mathrm{sMAPE} = \frac{\overline{2|y - \widehat{y}|}}{|y| + |\widehat{y}|}, \tag{21}$$

each averaged within a configuration and then divided by the Seasonal Naive value for that configuration before the geometric mean over configurations defined in the main paper is taken. Aggregation is geometric rather than arithmetic throughout: the arithmetic mean of these ratios runs 5 to 22% higher because a few configurations dominate it. The ratios are well conditioned even where the underlying metric is not, which matters for MAPE, whose absolute Seasonal Naive value exceeds

Table 13: GIFT-Eval accuracy over 97 configurations, aggregate and by horizon group. Entries are normalized by Seasonal Naive within the corresponding group (55 short, 21 medium, 21 long); lower is better. Bold marks the best entry in each column.

		All		Short		Medium		Long	
Model	Setup	MASE	CRPS	MASE	CRPS	MASE	CRPS	MASE	CRPS
Seasonal Naive	Reference	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Sundial-Base-128M	Zero-shot	0.750	0.559	0.732	0.592	0.764	0.526	0.785	0.510
Moirai-2.0-R-Small	Pretrained	0.728	0.516	0.695	0.517	0.759	0.519	0.789	0.513
TimesFM-2.5	Zero-shot	0.705	0.490	0.681	0.504	0.724	0.472	0.751	0.475
Granite-FlowState-r1.1	Zero-shot	0.701	0.490	0.680	0.507	0.722	0.474	0.739	0.464
Chronos-2	Pretrained	0.698	0.485	0.667	0.496	0.725	0.471	0.757	0.472
TiRex-2-Zeroshot	Zero-shot	0.697	0.478	0.679	0.500	0.704	0.450	0.739	0.451
Toto-2.0-2.5B	Pretrained	0.696	0.476	0.663	0.486	0.726	0.464	0.754	0.461
Timer-S1	Pretrained	0.693	0.485	0.683	0.499	0.705	0.469	0.709	0.467
MORPHEUS-2.0	Pretrained	0.689	0.522	0.747	0.613	0.633	0.446	0.606	0.402
Falcon-X	Pretrained	0.687	0.486	0.704	0.529	0.664	0.437	0.668	0.431
Granite-PatchTST-FM-r2	Zero-shot	0.685	0.467	0.665	0.489	0.700	0.439	0.723	0.441
Toto-2.0-2.5B-FT	Fine-tuned	0.679	0.463	0.653	0.476	0.706	0.450	0.724	0.444
TiRex-2-Pretrained	Pretrained	0.678	0.467	0.662	0.489	0.692	0.443	0.706	0.436
Toto-2.0-FnF	Pretrained	0.676	0.463	0.650	0.479	0.707	0.449	0.717	0.439
EXAONE-Forecast	Zero-shot	0.673	0.460	0.660	0.484	0.680	0.430	0.702	0.431
TimesFM-3	Zero-shot	0.667	**0.456**	**0.648**	**0.475**	0.685	0.432	0.700	0.430
Falcon-2.0	Pretrained	0.666	0.486	0.652	0.492	0.667	0.474	0.703	0.483
MORPHEUS-MULTISCALE	Pretrained	**0.653**	0.476	0.689	0.538	**0.622**	**0.427**	**0.598**	**0.385**

10^{13} on configurations with near-zero targets while its ratio stays inside $[0.03, 2.60]$ across every model.

Table 14 reports the result. MORPHEUS-MULTISCALE leads on MASE, ND, MSE, and MAPE, so its advantage is not an artifact of the seasonal scaling that MASE applies: ND normalizes by total label mass instead and MSE not at all, and the ordering is unchanged under both. TimesFM-3, the nearest competitor in aggregate, leads the remaining two columns, sMAPE and CRPS. This ordering reflects the composition of the benchmark. Of the 97 configurations 55 are short-horizon, so every column of Table 14 is weighted 57% toward short requests, and decomposing the same six metrics by horizon group separates the two systems: TimesFM-3 is ahead on all six at short horizons, whereas MORPHEUS-MULTISCALE is ahead on five of six at medium and on all six at long, where its relative MSE is 0.414 against 0.536. The two models are strong in different regimes, and the aggregate reports the regime that occupies the majority of the benchmark.

E3 ACCURACY BY DOMAIN

This section decomposes the aggregate of Section 6 of the main paper by subject matter. Table 14 aggregates over configurations that differ widely in subject matter. Table 15 decomposes the two headline metrics into the seven GIFT-Eval domains, normalizing within each domain by Seasonal Naive exactly as before.

Among the systems compared, MORPHEUS-MULTISCALE leads relative MASE on Energy, the largest domain, and on Transport, which together hold 47 of the 97 configurations; it stays within 2–5% of the best system on Web/CloudOps, Nature, and Sales, and trails on Econ/Fin and Healthcare. The winning domains are the ones with many correlated variables sampled frequently enough that several patch widths describe genuinely different structure, the regime the mixing head targets. The losing domains are short, sparse, and low-frequency, where the widest encoder sees little history and the narrowest sees mostly noise. The same split is sharper on CRPS, where our margin survives only on Energy and Transport. The probabilistic gap is not spread evenly: it concentrates in Healthcare, Web/CloudOps, and Nature, where both Chronos-2 and TimesFM-3 are ahead of us.

Table 14: All distinct GIFT-Eval metrics over 97 configurations, each normalized by Seasonal Naive and aggregated as a geometric mean; lower is better. Relative MAE, RMSE and NRMSE are omitted because they are determined by ND and MSE. Bold marks the best entry in each column.

Model	MASE	ND	MSE	MAPE	sMAPE	CRPS
Seasonal Naive	1.000	1.000	1.000	1.000	1.000	1.000
Sundial-Base-128M	0.750	0.739	0.546	0.777	0.945	0.559
Moirai-2.0-R-Small	0.728	0.705	0.545	0.693	0.925	0.516
TimesFM-2.5	0.705	0.687	0.482	0.707	0.856	0.490
Granite-FlowState-r1.1	0.701	0.681	0.493	0.671	0.833	0.490
Chronos-2	0.698	0.679	0.495	0.666	0.886	0.485
TiRex-2-Zeroshot	0.697	0.673	0.483	0.641	0.888	0.478
Toto-2.0-2.5B	0.696	0.670	0.499	0.655	0.885	0.476
Timer-S1	0.693	0.668	0.484	0.663	0.890	0.485
MORPHEUS-2.0	0.689	0.675	0.489	0.626	0.897	0.522
Falcon-X	0.687	0.670	0.476	0.655	0.886	0.486
Granite-PatchTST-FM-r2	0.685	0.661	0.475	0.658	0.876	0.467
Toto-2.0-2.5B-FT	0.679	0.655	0.480	0.642	0.867	0.463
TiRex-2-Pretrained	0.678	0.659	0.466	0.651	0.868	0.467
Toto-2.0-FnF	0.676	0.653	0.463	0.645	0.867	0.463
EXAONE-Forecast	0.673	0.650	0.449	0.646	0.876	0.460
TimesFM-3	0.667	0.643	0.444	0.628	**0.796**	**0.456**
Falcon-2.0	0.666	0.648	0.462	0.636	0.861	0.486
MORPHEUS-MULTISCALE	**0.653**	**0.642**	**0.443**	**0.603**	0.855	0.476

Table 15: GIFT-Eval accuracy by domain over 97 configurations. Entries are geometric means of the per-configuration ratio to Seasonal Naive within the domain; lower is better. Bold marks the best entry in each column within each metric.

Model	Energy (32)	Web/Cloud (20)	Nature (15)	Transport (15)	Econ/Fin (6)	Health (5)	Sales (4)	All (97)
Relative MASE								
TimesFM-3	0.798	**0.582**	**0.687**	0.543	**0.739**	**0.535**	0.682	0.667
Chronos-2	0.816	0.611	0.723	0.603	0.772	0.551	**0.679**	0.698
MORPHEUS-2.0	0.755	0.614	0.746	0.560	0.853	0.640	0.759	0.689
MORPHEUS-MULTISCALE	**0.706**	0.596	0.719	**0.533**	0.791	0.595	0.705	**0.653**
Relative CRPS								
TimesFM-3	0.567	**0.418**	**0.326**	0.402	**0.717**	**0.400**	0.411	**0.456**
Chronos-2	0.585	0.469	0.340	0.451	0.735	0.423	**0.406**	0.485
MORPHEUS-2.0	0.578	0.527	0.401	0.441	0.840	0.566	0.494	0.522
MORPHEUS-MULTISCALE	**0.519**	0.493	0.372	**0.401**	0.774	0.503	0.430	0.476

E4 CALIBRATION OF THE QUANTILE GRID

This section supports the interval-width claim of Section 8 of the main paper. Relative CRPS scores sharpness and calibration together, so it cannot say whether the emitted intervals are the right width. We compare the empirical frequency $\Pr(y \le \widehat{q}_\ell)$ against the nominal level ℓ at each of the 21 emitted quantiles, macro-averaged over the 97 configurations (Table 16).

Every level below the median over-covers and every level above it under-covers, so the predictive distribution is too narrow rather than displaced: the nominal 80% interval of MORPHEUS-2.0 covers under half of the observations. Multiresolution fusion therefore also improves calibration, which the accuracy tables do not show, since averaging the components' quantile functions widens the grid relative to the standalone model. It lifts that interval by about ten points and cuts the largest deviation from nominal by nearly a third, though the intervals stay too narrow. What remains grows with the requested horizon and is worst at long, consistent with median feedback carrying no uncertainty forward across rollout steps. Quantile crossing is immaterial by comparison, affecting roughly one adjacent pair in a hundred at a mean violation of 0.3% of the grid span.

Table 16: Calibration of the emitted quantile grid over 97 configurations. Interval columns give the empirical coverage of the central interval at the stated nominal level; the nominal value is the target. Max. deviation is the largest gap between empirical and nominal coverage over the 21 emitted levels. Rescaled rows widen the grid about its own median by one fixed factor, $\widehat{q}_\ell \mapsto \widehat{q}_{0.5} + \eta(\widehat{q}_\ell - \widehat{q}_{0.5})$ with $\eta = 1.4$, which leaves every forecast median and therefore MASE unchanged. The factor was derived on the GIFT-Eval training split; it is a post-hoc diagnostic and is applied to no reported result.

	Central interval coverage			
System	60%	80%	95%	Max. dev.
MORPHEUS-2.0	32.0	47.9	69.1	0.178
rescaled, $\eta = 1.4$	42.7	58.7	77.7	0.122
MORPHEUS-MULTISCALE	41.3	57.7	79.5	0.127
rescaled, $\eta = 1.4$	53.3	69.1	86.2	**0.069**

The defect is a width rather than a location, so the predictive spread is the most direct route to the remaining probabilistic gap, whether by scoring a continuous ranked criterion rather than a finite pinball sum or by tying the grid to a parameterized width. Both can be applied without changing the architecture.