FRACAST-0: FRACTAL WEIGHT SHARING FOR A TIME SERIES FOUNDATION MODEL WITH ONLY 85K PARAMETERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series foundation models must preserve multi-domain breadth, probabilistic output, and multiple temporal scales, but parameter count grows when each scale receives a separate representation. We introduce Fracast-0, a probabilistic forecasting foundation model that exploits temporal self-similarity to reuse one operator across scales. A parameter-free detector extracts significant seasonal structure. The encoder applies a shared local block along a geometric dilation ladder with scale conditioning, while the decoder combines context-gathered states with an explicit seasonal future state and reuses a second block along another ladder before emitting nine quantiles. Pretraining across six corpora preserves multi-domain breadth within 85,001 parameters. On 97 GIFT-Eval configurations without per-dataset fine-tuning, Fracast-0 is the smallest of 28 evaluated checkpoints and remains non-dominated in the aggregate parameter-accuracy plane with MASE 0.808 and WQL 0.564. It uses 42.0% fewer parameters than TinyCast, whose MASE and WQL are 4.2% and 3.3% lower. These results support cross-scale weight reuse as a practical route to further time series foundation model compression. The official repository is released at `https://anonymous.4open.science/r/fracast-0`.

1 INTRODUCTION

Time series foundation models share a simple premise. Pretraining across domains can produce forecasts without fitting each dataset. The earliest systems put that premise into different shapes, with TimesFM using a patched decoder, Moirai handling arbitrary variates and horizons with a masked encoder, and Chronos tokenizing numeric values for language-model-style pretraining (Das et al., 2024; Woo et al., 2024; Ansari et al., 2024). The design space then widened as YingLong emphasized output scaling, TiRex and FlowState changed how temporal information is propagated, and Chronos-2 and Moirai-2.0 extended multivariate and decoder-only forecasting (Wang et al., 2025b; Auer et al., 2025; Graf et al., 2026; Ansari et al., 2025; Liu et al., 2025). Recent systems made these choices sharper still. Toto-2.0 examined what scaling buys, TiRex-2 carried recurrent forecasting into streaming, and Falcon-2.0 made heterogeneous pretraining regimes explicit (Khwaja et al., 2026; Podest et al., 2026; Xia et al., 2026). Against this increasingly specialized design space, TinyCast compressed a probabilistic zero-shot forecaster into the hundred-thousand-parameter range (Steinhauser, 2026). GIFT-Eval provides the common evaluation surface for these choices, covering 23 datasets and short, medium, and long horizons without per-dataset fine-tuning (Aksu et al., 2024).

These developments turn capacity from a background assumption into an explicit design choice. Within the compared checkpoints, TinyCast alone falls below one million parameters, using a 146K-parameter budget to compute periodic structure rather than store a large backbone. This establishes the possibility without determining a floor. Pushing below this range changes the design problem. **Breadth, probabilistic output, and multiple temporal scales must be preserved without letting parameter count grow with the number of scales.** (Wang et al., 2025a) *Can a general probabilistic forecaster retain multi-domain evidence at a still smaller scale by reusing one operator across scales instead of assigning separate weights to each?* The two parameter layouts in Figure 1 make this design choice concrete. This paper pursues that question.

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

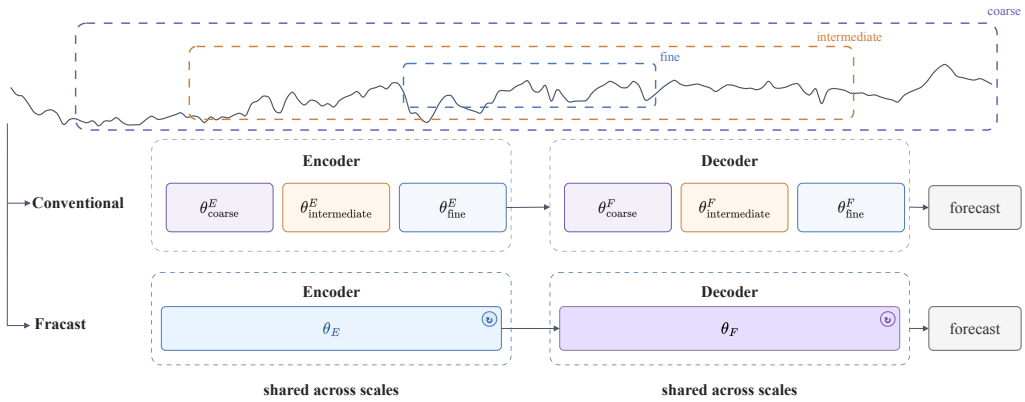


Figure 1: Cross-scale parameter growth. A conventional multiscale design assigns a separate encoder and decoder parameter set at every temporal scale. Fracast-0 instead reuses one encoder operator and one decoder operator across scales, so the stored parameter count does not grow with the number of scale rungs.

Fracast-0 answers by treating temporal self-similarity as a **parameter-sharing principle**. A parameter-free detector extracts significant seasonal structure from the input. The encoder then applies one local block along a geometric dilation ladder. Scale conditioning tells the shared block which dilation it occupies, so scale changes computation rather than weights. The decoder combines context-gathered states with a seasonal future state, reuses a second local block along another dilation ladder, and emits nine quantiles. Repeated operators therefore provide multi-scale reach within an 85K-parameter budget, while pretraining across six corpora supplies breadth.

On 97 GIFT-Eval configurations without per-dataset fine-tuning, Fracast-0 is the smallest of 28 evaluated checkpoints and remains **non-dominated** in the aggregate parameter-accuracy plane with MASE 0.808 and WQL 0.564 (Aksu et al., 2024). TinyCast uses 42.0% more parameters and improves both aggregate metrics by 4.2% and 3.3%, while the Fracast-0 gap widens from short to long horizons. Together, these results support **cross-scale weight reuse** as one practical route to a smaller general forecaster while preserving the breadth, quantile output, and multi-scale setting studied here, rather than as a universal advantage of small models.

2 Related Works

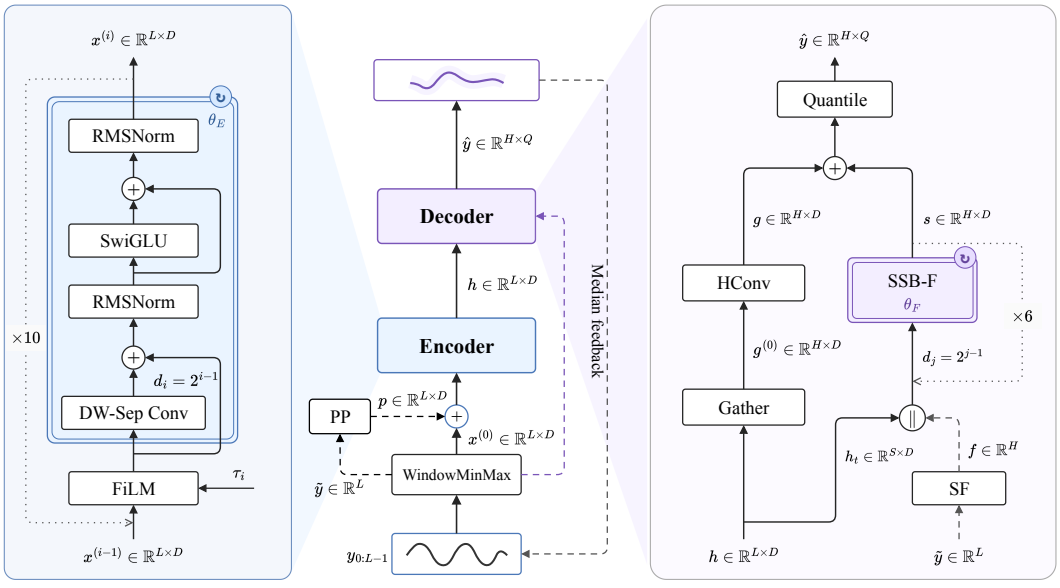
Self-similar and multiscale models. Self-similarity lets one local rule recur at different scales. Mandelbrot framed natural length as scale-dependent. In deep learning, FractalNet uses a recursive topology, WaveNet expands receptive fields through dilation, and ALBERT ties parameters across depth (Mandelbrot, 1967; Larsson et al., 2017; van den Oord et al., 2016; Lan et al., 2020). Time series models adapt this principle in different ways. TimesNet aligns multi-periodic variation in 2D tensors, TimeMixer mixes coarse and fine components, and FlowState remains equivariant to sampling-rate changes (Wu et al., 2023; Wang et al., 2024; Graf et al., 2026). Scaleformer is the closest prior method to Fracast-0. It iteratively refines predictions with *one set of weights shared across scales* (Shabani et al., 2023). Fracast-0 shares parameters **inside the forecasting block itself**. The encoder and future-state decoder each reuse one local update along their dilation ladders, while scale conditioning separates the role of each rung.

Time series foundation models and GIFT-Eval. Scale sharing is only one part of a foundation-model design. Such a model must also decide how to pretrain across domains and how to represent different variates, frequencies, and horizons. The first designs already disagreed about the representation. TimesFM used a patched decoder, Chronos tokenized numeric values for probabilistic decoding, and Moirai treated arbitrary variates and horizons with a universal transformer (Das et al.,

2024; Ansari et al., 2024; Woo et al., 2024). Later work extended this setup toward scaling, recurrent state, streaming, and multivariate inference (Khwaja et al., 2026; Auer et al., 2025; Podest et al., 2026; Ansari et al., 2025; Liu et al., 2025). GIFT-Eval includes earlier probabilistic and pretrained designs such as Lag-Llama, Timer, and TTM (Rasul et al., 2023; Liu et al., 2024; Ekambaram et al., 2024). Reverso and KAIROS target compact or adaptive modeling (Fu et al., 2026; Ding et al., 2025). YingLong and Falcon-2.0 explore output scaling and heterogeneous training regimes (Wang et al., 2025b; Xia et al., 2026). TinyCast instead pushes the same setup toward a much smaller backbone (Steinhauser, 2026). The benchmark puts these choices on one evaluation surface with 23 datasets across domains and sampling frequencies. For each short-, medium-, or long-horizon configuration, it reports MASE and WQL relative to Seasonal Naive, and its protocol uses no per-dataset fine-tuning (Aksu et al., 2024). Under this protocol, accuracy is easier to interpret when read alongside model size. We therefore place Fracast-0 at the 85K point on the parameter-accuracy frontier.

3 METHODOLOGY

Fracast-0 processes a univariate context window in three stages, illustrated in Figure 2. Channels are forecast independently with the same parameters, so a multivariate series needs no change to the architecture. The preprocessing stage normalizes the raw context and derives its periodic prior. The encoder maps the normalized context to a hidden state. The decoder maps that state to quantile forecasts over the target horizon.



(a) Encoder (b) Overall (c) Decoder

Figure 2: Fracast-0 turns a univariate context window into a quantile forecast through three stages. A preprocessing stage normalizes the raw context and derives its periodic prior. An encoder maps the normalized context to a hidden state, and a decoder maps that state back to the horizon. Panel (b) shows the overall path. Panels (a) and (c) expand the encoder and the decoder.

3.1 PREPROCESSING

The preprocessing stage turns the raw context window into the initial state $x^{(0)} \in \mathbb{R}^{L\times D}$ that the encoder consumes. Its input is a univariate window $y \in \mathbb{R}^{L}$ together with a binary mask $m \in \{0,1\}^{L}$, where $m_t = 1$ marks a position that carries a measurement and $m_t = 0$ marks a gap.

WindowMinMax. Series arrive at arbitrary and unrelated scales. Each window is normalized by its own range in Equation 1; this is Fracast-0's WindowMinMax transform. RevIN (Kim et al., 2022) motivates reversible per-window normalization but uses a different mean/std rule. Let $v_{\min} =$

$\min_t y_t$ and $v_{\max} = \max_t y_t$ be the window minimum and maximum, and let ϵ keep the denominator away from zero. The resulting $\tilde{y} \in \mathbb{R}^L$ lies in $[0, 1]$. The two statistics are computed once per window and receive no gradient, and they are reused to invert the forecast at inference. Non-finite values are replaced by zero, and positions marked as missing in m are set to zero after normalization. The same $\tilde{y}$ also feeds the seasonal fill of the decoder.

$$\tilde{y}_t = \frac{y_t - v_{\min}}{\max\left(v_{\max} - v_{\min},\, \epsilon\right)}, \tag{1}$$

Input projection. Each position is described by seven features. The first is the normalized value $\tilde{y}_t$. The second is a coverage channel c_t that repeats the observation mask as a numeric feature, so that a missing position stays distinguishable from a measured zero. The remaining five form a bounded recency basis r_t, which records how far the position lies from the end of the window. Two of its channels carry the signed distance $\delta_t = (t - (L-1))/L$ linearly and logarithmically, and three are exponential decays of different rates. A single linear layer maps the concatenation $u_t = [\tilde{y}_t, c_t, r_t] \in \mathbb{R}^7$ to the model width D.

Periodic prior. The seasonality of an unseen series has to be read from the context itself. Fracast-0 uses a parameter-free detector that scores local peaks of the normalized periodogram under a Bonferroni-corrected threshold and keeps the K most significant periods (Fisher, 1929), with $K = 4$ in this work; the Fisher result motivates the extremal-threshold idea rather than defining the implemented test. Every retained period P_k contributes a sine and a cosine channel at each position, $\sin(2\pi t/P_k)$ and $\cos(2\pi t/P_k)$, and periods rejected by the significance test leave their channels at zero. A linear layer projects these $2K$ channels to D dimensions and produces the periodic prior (PP) $p \in \mathbb{R}^{L\times D}$. The preprocessing output is the sum of the projected features and the prior, $x^{(0)} = \mathrm{Linear}(u) + p$, which is the input of the first encoder stage.

3.2 Encoder

The encoder maps the projected context $x^{(0)} \in \mathbb{R}^{L\times D}$ to a hidden state $h \in \mathbb{R}^{L\times D}$ that covers the whole window, and Panel (a) of Figure 2 shows its layout. A time series often repeats its local shape at several scales, the same statistical self-similarity that makes the measured length of a coastline depend on the ruler (Mandelbrot, 1967). This self-similar structure is the fractal property in the name of the model, and Fracast-0 takes it as the inductive bias of the encoder. A geometric dilation ladder gives the encoder reach over the window without any global mixing operator, and one block runs at every rung of the ladder, so the same local update applies at every scale and the parameter count does not grow with the number of scales. Fracast-0 follows the cross-layer parameter sharing of ALBERT (Lan et al., 2020) and **ties the entire block across all ten dilations**.

Scale conditioning. One block sees ten different dilations $d_i = 2^{i-1}$, so the scale has to enter through its input rather than through separate weights. Fracast-0 conditions the block on the continuous coordinate $\tau_i = \log_2 d_i$ with the feature-wise affine transform of FiLM (Perez et al., 2018) in Equation 2,

$$\begin{aligned} &\phi_m(\tau_i) = \sin(\pi\tau_i/2^m), \quad \phi_{M+m}(\tau_i) = \cos(\pi\tau_i/2^m), \quad m = 0, \ldots, M-1, \\ &[\gamma(\tau_i); \beta(\tau_i)] = W_s\phi(\tau_i) + b_s, \\ &\mathrm{FiLM}(x, \tau_i) = x \odot \big(1 + \gamma(\tau_i)\big) + \beta(\tau_i), \end{aligned} \tag{2}$$

where $M = 4$ in this work, $\phi(\tau_i) \in \mathbb{R}^{2M}$ is the fixed sine/cosine scale feature (Vaswani et al., 2017), and $W_s \in \mathbb{R}^{2D\times 2M}$ and $b_s \in \mathbb{R}^{2D}$ are initialized to zero. Conditioning therefore starts as the identity, and the shared block can fit all scales before any per-scale adjustment grows in.

Shared block. Stage i updates the state in Equation 3. One set of weights θ_E serves the whole recurrence, so the ladder is a single local operator evaluated at ten scales.

$$\begin{aligned} a^{(i)} &= \mathrm{FiLM}\big(x^{(i-1)}, \tau_i\big), \\ z^{(i)} &= \mathrm{RMSNorm}\Big(a^{(i)} + \mathrm{DSConv}_{d_i}\big(a^{(i)}\big)\Big), \\ x^{(i)} &= \mathrm{RMSNorm}\Big(z^{(i)} + \mathrm{SwiGLU}\big(z^{(i)}\big)\Big), \end{aligned} \tag{3}$$

where $i = 1, \ldots, N$, and DSConv_{d_i} is the causal depthwise-separable convolution (DW-Sep Conv) (Howard et al., 2017) of kernel size three whose dilation follows the ladder of WaveNet (van den Oord et al., 2016). The depthwise stage filters every channel along time on its own, and the pointwise stage mixes channels. Both residual sums are normalized with RMSNorm (Zhang & Sennrich, 2019). The feed-forward path is a SwiGLU block (Shazeer, 2020), and the whole block holds 23,232 parameters. Preprocessing contributes 1,088 parameters, while FiLM, the shared block, and the final RMSNorm contribute 24,448 to the encoder, so the full core holds 25,536 parameters. This final RMSNorm turns $x^{(N)}$ into the hidden state h.

3.3 Decoder

The decoder maps $h \in \mathbb{R}^{L \times D}$ to $\hat{y} \in \mathbb{R}^{H \times Q}$ for $H = 48$ steps and $Q = 9$ quantile levels; Panel (c) of Figure 2 shows its layout. The gather state g summarizes the encoded context at every future position, while the future-state residual s evolves causally from a seasonal reference. Their sum determines the quantiles.

Gather path. This path reduces the encoded context to a shared summary, gives each horizon position an explicit query, and refines the positions causally together. Let $\bar{h} = L^{-1} \sum_{\ell=1}^{L} h_\ell$ and $c = [\bar{h}; h_L] \in \mathbb{R}^{2D}$. For $j = L, \ldots, L + H - 1$, the joint projection and horizon refinement are

$$z_j = W_g[r_j; c] + b_g + u_j, \qquad g_j = \mathcal{H}_2\big(\mathcal{H}_1(\mathcal{S}(z_j))\big), \tag{4}$$

where $r_j \in \mathbb{R}^5$ is the bounded recency feature, $u_j \in \mathbb{R}^D$ is the learned position query, and $W_g \in \mathbb{R}^{D \times (2D+5)}$ and b_g form a linear layer. The normalized feed-forward update and horizon refinement map are

$$\mathcal{S}(z) = \mathrm{RMSNorm}\big(z + \mathrm{SwiGLU}(z)\big), \tag{5}$$

$$\mathcal{H}_\ell(z) = \mathrm{RMSNorm}\big(z + \mathrm{Conv}_{2^{\ell-1}}(z)\big), \quad \ell = 1, 2, \tag{6}$$

where each convolution is causal, depthwise, and of kernel size three. The output is $g \in \mathbb{R}^{H \times D}$.

Future-state path. The gather path carries context only through a pooled summary, so this path injects an explicit seasonal state and evolves it causally over the horizon. The detector chooses the primary period P_1 among up to $K = 4$ significant normalized-periodogram candidates, or $P_1 = 1$ when none is significant. Each future position t enters one of $B = 16$ phase bins $b(t) = \lfloor B(t \bmod P_1)/P_1 \rfloor$; the seasonal fill in that bin is the mean of observed normalized context values. An empty bin uses the context mean, so the fill f_j is deterministic. It is projected with recency, $p_j = W_f[f_j; r_j] + b_f$, and the shared state begins as $z_F^{(0)} = [h_{L-S+1:L}; p_1; \ldots; p_H]$, where $S = 128$. Let $\mathcal{F}_6$ denote six applications of the decoder-specific SSB-F block with dilations $1, 2, 4, 8, 16, 32$. Each application composes a causal kernel-three depthwise-separable convolution, residual RMSNorm, SwiGLU, and RMSNorm, without FiLM. The residual state is

$$s = W_F\big[\mathcal{F}_6\big(z_F^{(0)}\big)\big]_{S+1:S+H} + b_F, \qquad W_F = 0,\ b_F = 0, \tag{7}$$

where only the final H states are read and $s \in \mathbb{R}^{H \times D}$; zero initialization makes the path an identity residual at the start of training.

Combination and rollout. The decoder computes $\hat{y} = W_o(g + s) + b_o$, where $W_o \in \mathbb{R}^{Q \times D}$ and $b_o \in \mathbb{R}^Q$. WindowMinMax statistics invert the normalized forecast at inference. Longer horizons feed the median forecast back into the context and rerun the same model without new parameters. The gather path with output projection holds 31,625 parameters, the future-state path holds 27,840, the decoder holds 59,465, and the model holds 85,001 in total.

3.4 TRAINING

Fracast-0 is trained on univariate windows in the normalized space. For normalized targets y_t, predicted quantiles $\hat{q}_t(\tau)$, observation mask o_t, and $\mathcal{Q} = \{0.1, 0.2, \ldots, 0.9\}$, the pinball loss of quantile regression (Koenker & Bassett, 1978; Das et al., 2024) is

$$\mathcal{L}_{\text{pin}} = \frac{1}{|\mathcal{Q}|} \sum_{\tau \in \mathcal{Q}} \frac{1}{\sum_{t=1}^{H} o_t} \sum_{t=1}^{H} o_t \max\left((\tau - 1)(y_t - \hat{q}_t(\tau)),\, \tau(y_t - \hat{q}_t(\tau))\right), \tag{8}$$

where positions with $o_t = 0$ and near-flat windows are excluded. The committing term uses the median error $m_t = |\hat{q}_t(0.5) - y_t|$ and the error $c_t = |s_t - y_t|$ of the seasonal-copy reference s_t. Let $\gamma = 1$ when $\sum_{t=1}^{H} o_t c_t < \sum_{t=1}^{H} o_t m_t$, and $\gamma = 0$ otherwise:

$$\mathcal{L}_{\text{commit}} = \gamma \, \frac{\sum_{t=1}^{H} o_t \max(0, m_t - c_t)}{\sum_{t=1}^{H} o_t}, \tag{9}$$

so each positive term penalizes a position where the median is less accurate than the seasonal copy, and γ disables the whole term when the copy is not better over the window. The total objective is

$$\mathcal{L} = \mathcal{L}_{\text{pin}} + \lambda \mathcal{L}_{\text{commit}}, \tag{10}$$

with $\lambda = 0.3$ reported in Appendix A. Training unrolls several future chunks and feeds each chunk with either the model median or the observed continuation under scheduled sampling that increases from zero, matching the objective to long-horizon rollout. Exact rollout length, sampling schedule, optimizer settings, and budget are reported in Appendix A.

4 EXPERIMENTS

Benchmark. We evaluate forecasting on GIFT-Eval (Aksu et al., 2024), a benchmark of 23 datasets across seven domains and ten sampling frequencies with short, medium, and long horizon terms, without per-dataset fine-tuning. We follow its official test protocol with the released windows and the nine quantile levels from 0.1 to 0.9. Point accuracy is the mean absolute scaled error (MASE) normalized by the Seasonal Naive score of the same configuration, and probabilistic accuracy is the weighted quantile loss (WQL); both are aggregated with a geometric mean over the 97 configurations, of which 55 are short, 21 medium, and 21 long. The scaling study compares released checkpoints of Moirai and Moirai 2.0 (Woo et al., 2024; Liu et al., 2025), TimesFM (Das et al., 2024), Chronos-Bolt and Chronos-2 (Ansari et al., 2024; 2025), Toto 2.0 (Khwaja et al., 2026), TiRex and TiRex-2 (Auer et al., 2025; Podest et al., 2026), FlowState (Graf et al., 2026), KAIROS (Ding et al., 2025), YingLong (Wang et al., 2025b), Falcon-2.0 (Xia et al., 2026), and TinyCast (Steinhauser, 2026); their scores are leaderboard entries recomputed under the same aggregation.

Pretraining data. Fracast-0 is pretrained on six public corpora: the GIFT-Eval pretraining corpus (Aksu et al., 2024), LOTSA, the Chronos datasets, BOOM, FEV, and the synthetic shards released with TinyCast (Steinhauser, 2026). No GIFT-Eval train split enters the corpus, and windows are drawn from all sources with a band-mix policy over sampling-rate bands.

Parameter budget. Fracast-0 reduces its budget by exploiting the self-similarity of temporal structure along the dilation ladder: one local block is reused at every scale, so the parameter count does not grow with the number of rungs. The largest single reduction is the periodic prior (-96.5%), while the main sharing gain is the encoder's dilated convolutions (-59.9%); the future-state path saves another 37.9%. These reductions outweigh the extra parameters in FiLM scale conditioning and the wider gather head. TinyCast (Steinhauser, 2026) provides a compact reference at this scale and shares the parameter-free detector, bounded recency basis, and seasonal fill. Table 1 compares the two budgets at the same context length $L = 2048$, block horizon $H = 48$, quantile count $Q = 9$ and width $D = 64$, and records where each model spends its budget rather than only what it saves.

4.1 FORECASTING ACCURACY

Performance and scaling. Table 2 gives the grouped scores and Figure 3 places them in the full released-checkpoint distribution. Fracast-0 obtains 0.808 overall MASE and 0.564 WQL with 85K

Table 1: TinyCast and Fracast-0 parameter budgets, counting shared tensors once.

Component	TinyCast	Fracast-0 (Ours)	Difference	Relative (%)
Input projection	960	512	-448	-46.7
Periodic prior	16,448	576	-15,872	-96.5
Scale conditioning	–	1,152	+1,152	–
Encoder, dilated convolutions	57,920	23,232	-34,688	-59.9
Encoder, final normalization	–	64	+64	–
Decoder, query and gather	26,313	31,625	+5,312	+20.2
Decoder, future-state path	44,864	27,840	-17,024	-37.9
Total	**146,505**	**85,001**	**-61,504**	**-42.0**

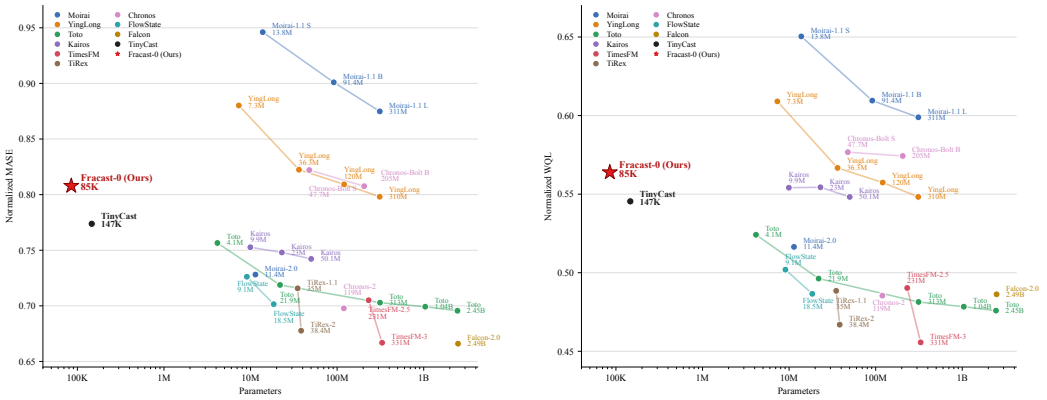
Table 2: GIFT-Eval test results without per-dataset fine-tuning. MASE and WQL are Seasonal Naive-normalized geometric means over 97 configurations (55 short, 21 medium, 21 long).

			Normalized MASE				Normalized WQL			
Model	Params	Pareto	Short	Medium	Long	Overall	Short	Medium	Long	Overall
Fracast-0 (Ours)	85K	**Both**	0.770	0.844	0.877	0.808	0.569	0.557	0.556	0.564
TinyCast	147K	**Both**	0.755	0.786	0.811	0.774	0.570	0.515	0.515	0.545
Moirai-2.0	11.4M	–	0.695	0.759	0.789	0.728	0.517	0.519	0.513	0.516
FlowState-r1.1	18.5M	**Both**	0.682	0.720	0.736	0.701	0.507	0.469	0.455	0.487
TiRex-2	38.4M	**Both**	0.662	0.692	0.706	0.678	0.489	0.443	0.436	0.467
Chronos-2	119M	–	0.667	0.725	0.757	0.698	0.496	0.471	0.472	0.485
YingLong-310M	310M	–	0.758	0.841	0.867	0.798	0.562	0.535	0.526	0.548
TimesFM-3	331M	**Both**	0.648	0.685	0.700	0.667	0.475	0.432	0.430	0.456
Toto-2.5B	2.45B	–	0.663	0.726	0.754	0.696	0.486	0.464	0.461	0.476
Falcon-2.0	2.49B	**MASE**	0.652	0.667	0.703	0.666	0.492	0.474	0.483	0.486

parameters, whereas TinyCast obtains 0.774 and 0.545 with 147K parameters. Thus the $1.72\times$ larger TinyCast is 4.2% better on MASE and 3.3% better on WQL. Table 2 also locates the WQL gap: the short-horizon values are nearly tied at 0.569 and 0.570, while the medium- and long-horizon values are 0.557 and 0.556 for Fracast-0 versus 0.515 and 0.515 for TinyCast. Figure 3(a) sharpens the scale contrast: FlowState-r1.1, TiRex-2, and TimesFM-3 reach 0.701, 0.678, and 0.667 at 18.5M, 38.4M, and 331M parameters, so these improvements over Fracast-0 cost $218\times$, $452\times$, and $3{,}891\times$ as many parameters. Figure 3(b) shows the same ordering on WQL (0.487, 0.467, and 0.456 versus 0.564), but the larger systems' gains over Fracast-0 are 13.7%, 17.2%, and 19.2%, far below their parameter multipliers. Table 2 and these two scaling panels therefore support the bounded conclusion that Fracast-0 is a **usable point near the extreme low-parameter end rather than the best absolute predictor**. The next development target is to close the medium- and long-horizon gap without surrendering the compact parameter budget.

Pareto optimality. Figure 4 isolates the empirical parameter–accuracy trade-off. In the overall panel, Fracast-0 is the first of the eight MASE-frontier steps at 85K parameters (0.808 MASE and 0.564 WQL). TinyCast is the next step: its extra 62K parameters buy 4.2% lower MASE and 3.3% lower WQL. The next MASE step, Toto-4M, requires another $28.3\times$ parameters for only 2.2% additional MASE improvement, and the final TimesFM-3 to Falcon-2.0 edge spends another $7.5\times$ parameters for 0.1%. In the horizon panel, the Fracast-0-to-TinyCast MASE changes are 2.0%, 7.4%, and 7.5% for short, medium, and long terms, showing that the smallest checkpoint is closest to its successor where prediction extends over fewer future blocks. The figure consequently establishes **non-domination and diminishing marginal returns** under this checkpoint set, but not a causal scaling law: the frontier mixes architectures, training corpora, and released schedules, and weight sharing itself remains untested without an unshared control at the same budget.

Horizon sensitivity. Figure 5 explains where the aggregate gap originates. In Figure 5(a), the short-horizon MASE values are 0.770 for Fracast-0 and 0.755 for TinyCast, only a 2.0% separation. Figure 5(b) moves to medium horizons, where the values become 0.844 and 0.786, widening the separation to 7.4%. Figure 5(c) gives long horizons at 0.877 and 0.811, a similarly clear 7.5% gap. From short to long horizons, Fracast-0's MASE increases by 13.9%, whereas TinyCast's increases



(a) Normalized MASE (b) Normalized WQL

Figure 3: Scaling study on GIFT-Eval. Every point is a released checkpoint, and lines connect checkpoints of the same family.

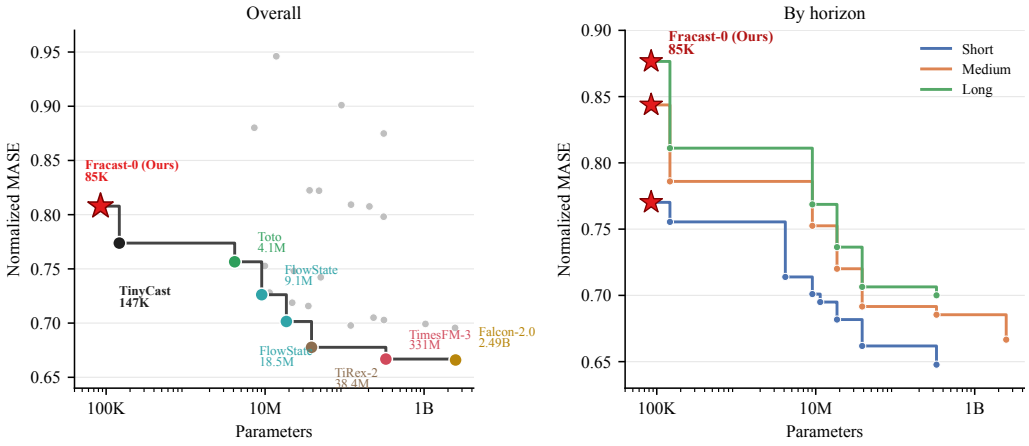


Figure 4: Parameter-accuracy Pareto frontier on GIFT-Eval.

by only 7.4%. This pattern is consistent with the cost of extending a 48-step block by repeated rollout: short targets need less error accumulation, whereas medium and long targets expose it. Because each horizon panel aggregates 55, 21, and 21 configurations, the figure does not isolate rollout length as the only cause. The resulting conclusion is **near short-horizon parity but weaker horizon stability beyond the short group**; the next diagnostic should evaluate medium- and long-horizon rollout before adding capacity.

4.2 Deployment efficiency

Performance test. CPU latency and resident memory were measured on a MacBook Air with an Apple M5 chip, 24 GB of unified memory, and 10 CPU cores, using PyTorch 2.14.0. All eight local checkpoints ran on CPU at batch size one under the same fixed random-input workload. Figure 6 tests whether the parameter reduction survives a fixed local batch-1 CPU workload. Figure 6a reports 23.9 ms for Fracast-0 and 41.4 ms for TinyCast, a 42.3% reduction that closely tracks the 42.0% parameter reduction. Figure 6b reports 389 versus 471 MiB but only a 17.4% reduction. This divergence indicates that runtime footprint is not parameter count alone because resident memory still depends on tensors, buffers, and framework overhead. The latency panel also bounds the claim: Toto-4M is fastest at 7.8 ms but uses 589 MiB, whereas TimesFM-3, TiRex-2, and FlowState-r1.1 are slower at 142.8, 224.4, and 256.2 ms and require 1,698, 1,058, and 2,431 MiB. Fracast-0 is therefore the **lightest resident-memory point and improves the compact TinyCast baseline on**

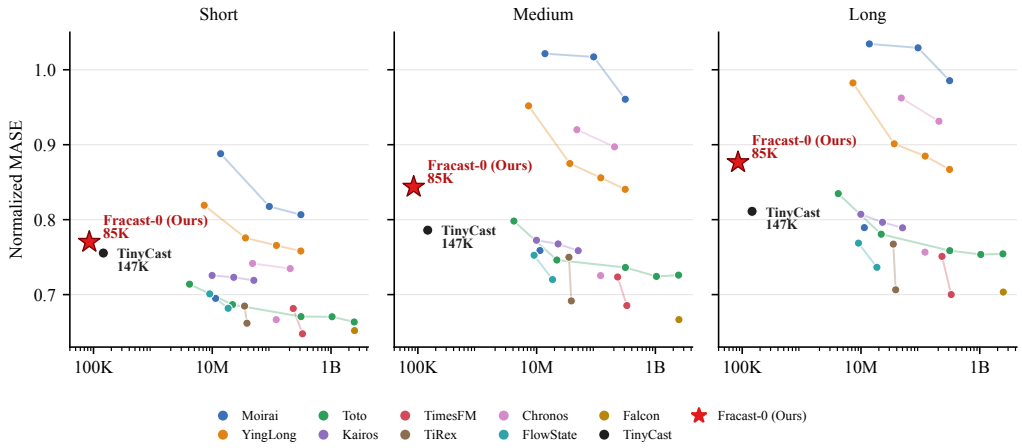
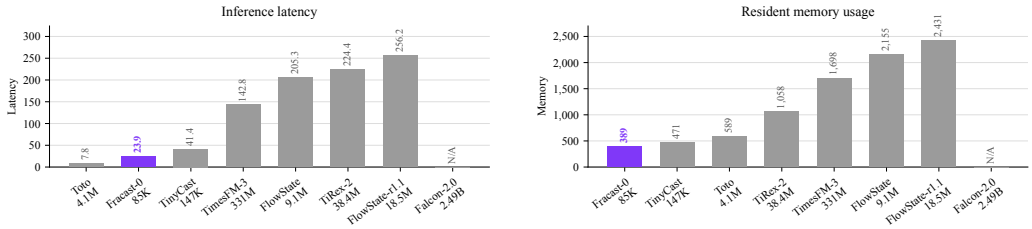


Figure 5: Normalized MASE against parameter count for the short, medium, and long horizon groups of GIFT-Eval.



(a) Batch-1 median latency (ms).

(b) Resident memory (MiB).

Figure 6: Deployment efficiency on the local CPU benchmark. Both panels use the same eight-model set and fixed random-input workload. The memory panel reports total resident memory after inference. Falcon-2.0 is N/A because no local checkpoint is available.

both CPU metrics, but it is not a universal fastest model. Deployment claims should remain within this local batch-1 workload until percentile latency and peak-memory distributions are reported.

Deployment implications. The short-horizon parity and CPU profile favor Fracast-0 when many local series must be scored on memory-constrained devices, especially at daily and hourly frequencies. Sub-hourly forecasting and extended rollouts remain where added capacity is more valuable, so deployment should weigh horizon, frequency, and local resource budget rather than parameter count alone. Appendix A.2 gives trajectory-level comparisons, while Appendices A.3–A.5 report domain, frequency, and variable-type breakdowns.

5 Conclusion

We presented Fracast-0, an 85K-parameter time series foundation model that exploits temporal self-similarity by reusing one block along each dilation ladder. Across 97 GIFT-Eval configurations, Fracast-0 remains non-dominated at MASE 0.808 and WQL 0.564, uses 42.0% fewer parameters than TinyCast, and has 42.3% lower batch-1 CPU latency. These results support cross-scale weight reuse as a viable path to further TSFM compression rather than establish it as a general law. Our evidence is limited to one Fracast-0 architecture and one benchmark, and we did not run large-scale ablations because compute was limited. Future work should test different stocks, horizons, and regimes at scale and determine which same-level weights to merge and which to keep separate, including whether depth still helps once weights are shared.

AI USE STATEMENT

In this work, we used generative AI tools, including large language models accessed through research, coding, and writing assistants, for literature discovery and summarization, brainstorming and refinement of research questions, conceptual framing, hypotheses, and claims, design and feedback on research methodology and experiments, implementation and debugging of model, training, data, and evaluation code, data preprocessing and translation or language editing, interpretation and presentation of experimental results, drafting and restructuring of manuscript sections, editing and polishing of manuscript prose, and preparation or modification of tables, figures, references, and other research artifacts.

We did not use generative AI tools to generate the training or evaluation datasets or to prove mathematical claims. We reviewed all AI-assisted work before using it in this paper. We checked literature claims and citations against primary papers and official repositories, verified the manuscript equations and technical descriptions manually, tested and inspected the code, and independently reproduced the reported numerical results from archived outputs. The authors made all final decisions about the research questions, method, experiments, claims, and presentation, reviewed every AI-assisted contribution.

REFERENCES

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. GIFT-Eval: A benchmark for general time series forecasting model evaluation, 2024. URL https://arxiv.org/abs/2410.10393.

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

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From univariate to universal forecasting, 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 38*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/5356603f9c47399adfd372f77a677057-Abstract-Conference.html.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/das24c.html.

Kuiye Ding, Fanda Fan, Zheya Wang, Hongxiao Li, Yifan Wang, Lei Wang, Chunjie Luo, and Jianfeng Zhan. KAIROS: Unified training for universal non-autoregressive time series forecasting, 2025. URL https://arxiv.org/abs/2510.02084.

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 37*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/874a4d89f2d04b4bcf9a2c19545cf040-Abstract-Conference.html.

Ronald Aylmer Fisher. Tests of significance in harmonic analysis, 1929. URL https://doi.org/10.1098/rspa.1929.0151.

Xinghong Fu, Yanhong Li, Georgios Papaioannou, and Yoon Kim. Reverso: Efficient time series foundation models for zero-shot forecasting, 2026. URL https://arxiv.org/abs/2602.17634.

Lars Graf, Thomas Ortner, Stanisław Woźniak, and Angeliki Pantazi. FlowState: Sampling-rate-equivariant time-series forecasting. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/65805.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications, 2017. URL https://arxiv.org/abs/1704.04861.

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

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cGDAkQo1C0p.

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

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020. URL https://iclr.cc/virtual/2020/poster/1935.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=S1VaB4cex.

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

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*, 2024. URL https://proceedings.mlr.press/v235/liu24cb.html.

Benoit B. Mandelbrot. How long is the coast of Britain? Statistical self-similarity and fractional dimension. *Science*, 156(3775):636–638, 1967. doi: 10.1126/science.156.3775.636. URL https://doi.org/10.1126/science.156.3775.636.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. doi: 10.1609/aaai.v32i1.11671. URL https://doi.org/10.1609/aaai.v32i1.11671.

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, 2026. URL https://arxiv.org/abs/2607.01204.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka, and Irina Rish. Lag-Llama: Towards foundation models for probabilistic time series forecasting, 2023. URL https://arxiv.org/abs/2310.08278.

Amin Shabani, Amir Abdi, Lili Meng, and Tristan Sylvain. Scaleformer: Iterative multi-scale refining transformers for time series forecasting. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sCrnllCtjoE.

Noam Shazeer. GLU variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Armin Steinhauser. TinyCast: Probabilistic zero-shot forecasting with computed periodicity. Weights: https://huggingface.co/raws-labs/tinycast, 2026. URL https://arxiv.org/abs/2608.15767.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A generative model for raw audio. In *9th ISCA Workshop on Speech Synthesis Workshop (SSW 9)*, 2016. URL https://www.isca-archive.org/ssw_2016/vandenoord16_ssw.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Mingxin Wang, Guohui Zhou, and Yong Deng. Uncertainty management with quantitative propensity matrix in random permutation set theory. *Information Sciences*, 692:121645, 2025a. URL https://doi.org/10.1016/j.ins.2024.121645.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7oLshfEIC2.

Xue Wang, Tian Zhou, Jinyang Gao, Bolin Ding, and Jingren Zhou. Output scaling: YingLong-delayed chain of thought in a large pretrained time series forecasting model, 2025b. URL https://arxiv.org/abs/2506.11029.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/woo24a.html.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Hongjie Xia, Yiding Liu, Yifan Hu, Peiyuan Liu, and Zewei Dong. Into the ORBIT for time series: Training regimes for foundation models, 2026. URL https://arxiv.org/abs/2608.13262. Falcon-2.0 technical report.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems 32*, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/1e8a19426224ca89e83cef47f1e7f53b-Abstract.html.

A Experiment Details

A.1 Training and Evaluation Protocol

Training configuration. Fracast-0 holds 85,001 parameters with width $D = 64$, ten dilations from 1 to 512, a context of 2,048 steps, a horizon of 48 steps, and nine quantiles. Pretraining uses AdamW with $\beta_2 = 0.95$, weight decay 0.01, and gradient clipping 1.0, with a peak learning rate of 3×10^{-3}, a 1,831-step warmup, a 65% stable fraction, and decay to 10^{-5}. The base schedule runs 36,621 steps at an effective batch of 4,096 windows (micro-batch 512 with 8 accumulation steps) under bf16 autocast, and the released checkpoint extends it to 40,075 steps, about 164M windows. The committing term carries $\lambda = 0.3$, training unrolls four future chunks with scheduled sampling rising to 0.5, and augmentation applies time flip, sign flip, down-sampling, and mixup.

Evaluation protocol. GIFT-Eval defines one evaluation unit by dataset, sampling frequency, and forecast term. This gives 97 released test configurations with 55 short, 21 medium, and 21 long members. We evaluate these windows without per-dataset fine-tuning and emit nine quantile levels from 0.1 to 0.9. For each unit, MASE is computed from the median forecast and WQL from the nine quantiles; both are divided by the corresponding Seasonal Naive score, so lower values are better. Overall and group scores are geometric means of these normalized ratios, which gives every configuration equal weight in the aggregate. Fracast-0 is evaluated with the local harness, while external per-configuration scores are recomputed from the GIFT-Eval Space CSVs[1].

A.2 Qualitative Short-Horizon Forecasts

Figure 7 compares the three models on six selected GIFT-Eval windows. The panels span smooth daily levels, sharp load changes, daily weather cycles, periodic solar generation, and noisy event counts. Fracast-0 has the lowest panel-title nMAE in all six windows, but the margins differ sharply. Electricity is the clearest case, where the relative gains over TinyCast and TimesFM-3 are 71.6% and 81.3%; KDD Cup is the closest case, where the corresponding gains are 5.3% and 2.7%. M4 Daily and KDD Cup have nearly overlapping medians, so their small score differences should not be read as visually decisive separations. In Electricity, the baselines introduce a large early peak, whereas Fracast-0 remains closer to the observed low level before tracking the later activity. ETTh1, Solar, and Jena Weather preserve the dominant periodic timing, and the remaining score differences mainly reflect peak height. Taken together, these panels indicate that Fracast-0's **clearest short-horizon advantage appears under abrupt load changes**, while its medians track the same dominant levels or cycles as the baselines elsewhere. The six cases therefore support a bounded conclusion that short-horizon tracking is competitive across these selected daily and hourly signals, but they do not estimate an aggregate win rate.

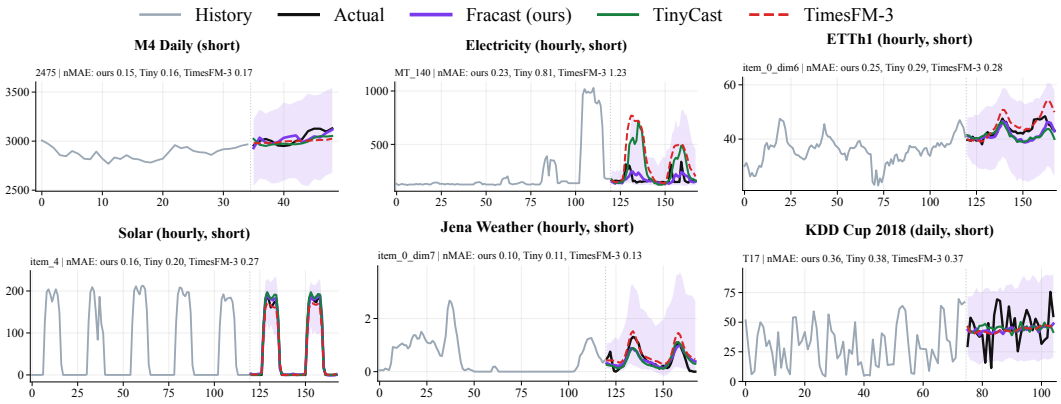


Figure 7: Qualitative short-horizon forecasts on six GIFT-Eval datasets. The shaded band is Fracast-0's 10–90% quantile interval, and the panel title reports normalized MAE for the selected window.

[1]Revision `81100d6fc0361dc011fb984d4e22d9b71af3c3c4`, accessed 2026-09-22.

A.3 OFFICIAL BY DOMAIN COMPARISON

Table 3 reports the seven-domain view of the ten checkpoints in Table 2 at GIFT-Eval Space. MASE is the Seasonal Naive-normalized geometric mean. Across domains, Fracast-0 remains below Seasonal Naive but **does not lead any domain**. TinyCast is better on six domains, with Sales as the exception, while TiRex-2, Chronos-2, TimesFM-3, Toto-2.5B, and Falcon-2.0 provide the strongest domain-level references.

Table 3: Official GIFT-Eval By Domain comparison. Each cell reports MASE.

Model	Econ/Fin	Energy	Healthcare	Nature	Sales	Transport	Web/CloudOps
Fracast-0 (Ours)	0.909	0.904	0.756	0.765	0.692	0.686	0.805
TinyCast	0.895	0.882	0.734	0.758	0.734	0.670	0.696
Moirai-2.0	0.779	0.837	0.600	0.755	0.689	0.620	0.665
FlowState-r1.1	0.756	0.816	0.597	0.705	0.682	0.604	0.628
TiRex-2	0.776	0.767	0.574	0.705	0.682	0.576	0.609
Chronos-2	0.772	0.816	0.551	0.723	0.679	0.603	0.611
YingLong-310M	0.899	0.870	0.703	0.746	0.742	0.666	0.846
TimesFM-3	0.739	0.798	0.535	0.687	0.682	0.543	0.582
Toto-2.5B	0.739	0.817	0.534	0.724	0.699	0.594	0.616
Falcon-2.0	0.694	0.808	0.563	0.640	0.688	0.575	0.575

A.4 OFFICIAL BY FREQUENCY COMPARISON

Table 4 reports the eight-frequency view of the ten checkpoints in Table 2. Fracast-0 remains below Seasonal Naive in seven groups; only Secondly is above 1.0, at 1.141. It tracks TinyCast within 1.4% on Daily, Hourly, and Weekly, is worse at Secondly and Minutely by 25.8% and 9.2%, and is better on Monthly, Quarterly, and Yearly by 3.4%, 6.8%, and 12.4%. The comparison therefore identifies **sub-hourly forecasting as the main frequency-specific weakness**, while the longer-period groups improve over TinyCast but remain behind TimesFM-3 and Falcon-2.0.

Table 4: Official GIFT-Eval By Frequency comparison. Each cell reports MASE.

Model	Secondly	Minutely	Hourly	Daily	Weekly	Monthly	Quarterly	Yearly
Fracast-0 (Ours)	1.141	0.826	0.765	0.711	0.869	0.855	0.801	0.818
TinyCast	0.907	0.756	0.754	0.711	0.859	0.885	0.859	0.934
Moirai-2.0	0.854	0.727	0.710	0.670	0.769	0.806	0.739	0.837
FlowState-r1.1	0.785	0.697	0.689	0.672	0.716	0.772	0.713	0.754
TiRex-2	0.721	0.684	0.631	0.654	0.770	0.784	0.737	0.892
Chronos-2	0.698	0.700	0.698	0.653	0.725	0.756	0.720	0.816
YingLong-310M	1.252	0.786	0.752	0.711	0.851	0.867	0.868	1.087
TimesFM-3	0.679	0.672	0.643	0.640	0.720	0.744	0.722	0.793
Toto-2.5B	0.760	0.686	0.682	0.655	0.751	0.775	0.719	0.867
Falcon-2.0	0.685	0.663	0.658	0.619	0.690	0.779	0.769	0.832

A.5 Official By Variate Type comparison

Table 5 reports the multivariate and univariate view of the same ten checkpoints. Fracast-0 obtains MASE 0.831 on multivariate inputs and 0.790 on univariate inputs, both below Seasonal Naive. The 0.041 absolute gap corresponds to a 5.2% relative increase, and every checkpoint in the table has higher multivariate MASE. Fracast-0 trails TinyCast in both groups and leads neither group, so its **remaining accuracy gap is broad rather than specific to one input type**, with multivariate inputs modestly harder.

Table 5: Official GIFT-Eval By Variate Type comparison. Each cell reports MASE.

Model	Multivariate	Univariate
Fracast-0 (Ours)	0.831	0.790
TinyCast	0.780	0.769
Moirai-2.0	0.752	0.709
FlowState-r1.1	0.717	0.689
TiRex-2	0.704	0.657
Chronos-2	0.709	0.689
YingLong-310M	0.840	0.766
TimesFM-3	0.679	0.658
Toto-2.5B	0.712	0.683
Falcon-2.0	0.683	0.653