LAGFORMER: REGRESSING PERIODS FROM SPARSE, IRREGULARLY SAMPLED SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Period estimation for irregularly sampled series is usually a search over trial frequencies, on a grid whose density grows with record length and whose highest peak can be a harmonic of the true period. Several learned estimators take such frequency-resolved inputs and inherit the same grid. We introduce LagFormer, which identifies the period from self-similarity on a logarithmic lag axis, where a fixed relative resolution suffices. An exact correlogram bins every pair of observations by time difference, handling arbitrary sampling without interpolation, and compares observed values alongside channels learned by a transformer. Since each period bin spans a whole number of lag bins, every candidate period's multiples lie at fixed lag offsets, which a harmonic comb gathers for a network outputting a distribution over log-period. A local refinement sharpens the estimate within the selected harmonic. Trained only on synthetic series that replay real sampling and noise, LagFormer finds the period within 10% on 90.2% of held-out synthetic series, against 69.3% for a Lomb–Scargle periodogram and 85.0% for PDM, without a fine global frequency search. On real light curves it agrees with the catalogue period on 80.6%, and its refined period folds the data more tightly than the catalogue period on 52.8% of a common same-harmonic subset, nearly the 53.7% from refining the catalogue period itself.

1 INTRODUCTION

Estimating the period of a recurring signal from irregularly sampled observations is usually posed as a search, in which a fit statistic is evaluated over a grid of trial frequencies and the best scoring point is returned (Lomb, 1976; Scargle, 1982; VanderPlas, 2018). The grid has to be finer than the response it is looking for, which has relative width about P/T for a period P over a record spanning a time T, so a longer record demands a denser grid. For the records used here that translates to about 10^6 trial frequencies per series. Density alone does not settle the question either, since the strongest peak can belong to a multiple of the true period (Graham et al., 2013a; VanderPlas, 2018).

Periodicity is a form of self-similarity, and a series that repeats with period P returns to the same phase at lags P, $2P$, $3P$ and further multiples. Asking where a series resembles itself is therefore an alternative to asking which frequency fits it, and it requires no trial frequencies. The difficulty is that the multiples which pin the period down lie far out, so the lag axis has to span the whole record while still resolving periods thousands of times shorter than it. Covering that range at fine resolution takes a large number of log-spaced lags, so the representation must store only a few values at each one. The correlogram below is built to meet this constraint.

Accordingly, a transformer encoder (Vaswani et al., 2017) turns each observation into a small multi-channel feature, with the normalised value kept as one of the channels, and self-similarity is measured between those features as a function of lag, so the cost of the axis is set by the number of channels and bins rather than by what is stored about each pair. On a logarithmic lag axis the multiples of a period are separated by fixed distances that do not depend on the period itself, so a single pattern of offsets, which we call the harmonic comb, reads the repeated structure of every period at once. A local refinement then sharpens the result.

We train on synthetic series that replay real sampling patterns and real noise with known periods, and evaluate in two settings. On synthetic data the period is known, so accuracy is measured directly. On real data we do not use the catalogue period as truth, and ask instead which period best explains

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.

the observations when the series is folded at it. LagFormer improves on the periodogram in both settings without a fine frequency search, and most remaining errors confuse a period with a small-integer multiple of it (Appendix D). The method uses no knowledge of what the series measures.

Our contributions are as follows.

- We formulate period estimation for irregularly sampled series as lag-domain inference, comparing learned per-observation features as a function of time lag, so that no fine trial-frequency grid is needed and the first stage never re-fits the observations at a trial period. Compressing the value axis into a small set of learned channels is what makes the lag axis long enough to carry the repeated structure the estimate relies on.
- We show that choosing the period bin width as an integer multiple of the lag bin width places every multiple of every period on the output grid at a constant offset, turning harmonic aggregation over exact lag statistics into a fixed gather with learned weights whose equal-weight special case aggregates value dispersion, akin to fold-based statistics.
- We evaluate on synthetic series with known periods and on real series, where the predicted period, the catalogue period and a classical periodogram are compared by how well each folds the observations. We report cost per series (Appendix A), the precision of the estimate inside its own output bin, and the structure of the errors that remain.

2 Related Work

Search over trial values. The standard estimators evaluate a fit statistic on a set of trial values and return the best scoring one. Least squares spectral analysis fits a sinusoid at each trial frequency (Lomb, 1976; Scargle, 1982; VanderPlas, 2018), and phase-binned methods fold at each trial period and score the folded values by their dispersion (Stellingwerf, 1978), an analysis of variance (Schwarzenberg-Czerny, 1989) or their conditional entropy (Graham et al., 2013b). Comparisons across these families (Graham et al., 2013a) differ in the statistic but share the commitment to a grid, so resolution is bought by grid density.

Learned and frequency-domain estimators. Networks have been trained to score or rank candidate periods (Estévez et al., 2010; Xu et al., 2022) and to regress the period directly (Claytor et al., 2022; Schochet et al., 2025). Both direct regressors take a frequency-resolved image of the series as input, a rebinned wavelet power spectrum in one case and a periodogram computed over successive chunks of the record in the other, so the trial grid moves into the input rather than disappearing. Other regressors read the series itself, Blancato et al. (2022) with a convolutional network and LightPred (Kamai & Perets, 2025) with the autocorrelation of the differenced series added, the input closest to ours, although that autocorrelation is a fixed function of the values where we correlate learned channels on an exact logarithmic lag axis and read them through a harmonic comb. Frequency-domain sequence models place a grid in frequency by construction (Zhou et al., 2022; Dong et al., 2025). Encoders built for irregular sampling (Che et al., 2018; Rubanova et al., 2019; Shukla & Marlin, 2021; Horn et al., 2020; Naul et al., 2018; Donoso-Oliva et al., 2023) target interpolation, classification and forecasting, and do not expose the period.

Lag-domain representations. RepNet (Dwibedi et al., 2020) and TransRAC (Hu et al., 2022) predict the period of a repeated action from the temporal self-similarity of video frames, and Autoformer recovers lag correlations through Fourier transforms (Wu et al., 2021), but all three assume uniformly sampled frames or sequences. Irregular series have been summarised by correlation functions that bin pairs of observations by lag (Edelson & Krolik, 1988; Rehfeld et al., 2011) and by histograms of pairwise time and value differences (Mahabal et al., 2017). We retain that pairwise view while learning the per-observation features, matching a logarithmic lag grid to the period grid and accumulating the statistics exactly without enumerating pairs.

3 Method

3.1 The problem, as the network sees it

The input is a series of n observations (t_i, m_i, e_i), where t_i is a timestamp, m_i is the observed value and e_i is its reported measurement error.

What makes the task hard is the sparsity of the sampling relative to the period. In the corpus used here a typical series spans a few thousand cycles of the signal but contains only about a thousand observations, so the typical series has less than one observation per cycle. Interpolating between consecutive observations cannot recover the waveform from so few, and evidence about the period has to come instead from comparing observations separated by many cycles.

The target is a single scalar $y = \log_{10} P$, and the model outputs a distribution over y on a uniform grid of $N_P = 512$ bins of width 0.008 over 0.01–120 time units, so that the resolution is uniform in relative terms. Figure 1 provides an overview of the first-stage estimator described below.

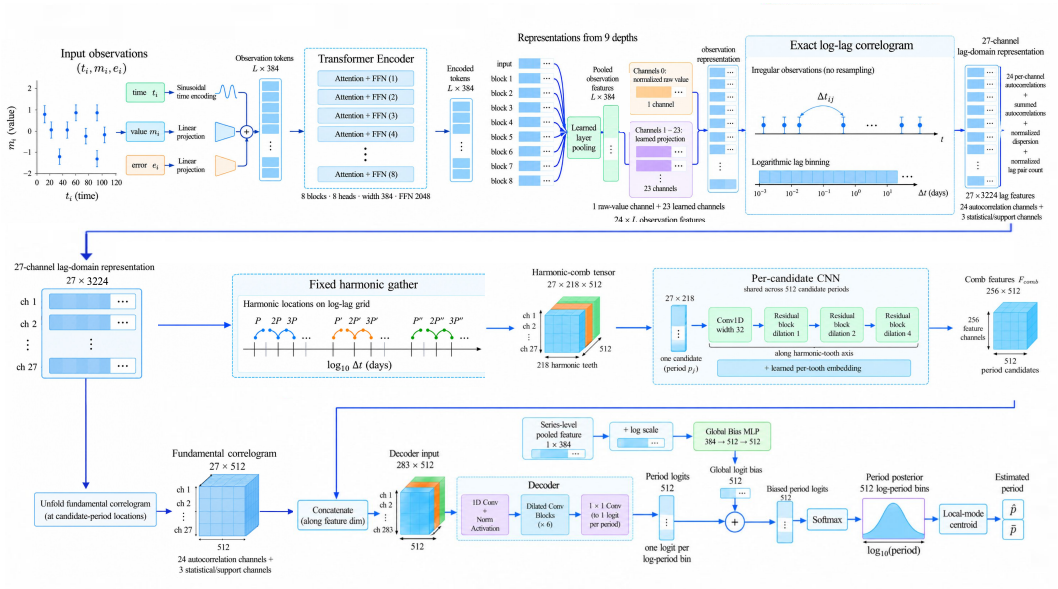

Figure 1: LagFormer first-stage architecture. A transformer encodes the irregular observations and pools representations across depth. An exact log-lag correlogram of the raw value and learned channels is gathered at fixed harmonic offsets for each output period bin. A shared comb network and decoder combine harmonic and fundamental-lag features to produce a posterior over log-period and a continuous estimate.

3.2 Normalisation and tokens

The model takes a fixed number of observations, $L = 1536$. A longer series, as 16.7 percent of the synthetic test series are, is restricted to a contiguous window of L of them, taken at a random offset in training and from the start of the record at evaluation, so its longest lag is set by that window rather than by the full record. Shorter series are padded, and padded entries enter no statistic.

Timestamps are made relative to the first observation in the window. Values are centred and divided by a per-series scale s, computed as a root mean square after clipping so that a few extreme points cannot set it, and measurement errors are divided by the same s, which places them on the same normalised scale. The scale itself is kept as $\log_{10} s$ and reaches the output only through Section 3.6. From here on, m_i and e_i denote the normalised value and error.

Each valid observation becomes one token

$$x_i = \mathrm{PE}(t_i) + W_m\, m_i + W_e \log e_i, \tag{1}$$

where PE encodes the timestamp with sinusoids (Vaswani et al., 2017; Donoso-Oliva et al., 2023) at $d/2$ log-spaced frequencies, with periods from 0.002 to 1000 time units, which locate each observation in time and are not searched over, and W_m and W_e are learned projections of the normalised value and the log of the normalised error. Temporal position is carried entirely by this encoding, with no learned position index.

3.3 ENCODER AND LAYER POOLING

The tokens pass through eight identical attention blocks of width $d = 384$, each with 8 heads and a feed-forward inner width of 2048, using post-norm residuals (Vaswani et al., 2017) and layer normalisation (Ba et al., 2016),

$$x \leftarrow \mathrm{LN}\big(x + \mathrm{Attn}(x)\big), \qquad x \leftarrow \mathrm{LN}\big(x + \mathrm{FFN}(x)\big). \tag{2}$$

Attention is full, and no relative-time bias is added since the next stage measures lag dependence explicitly. The depth at which features are most comparable across pairs of observations is not known in advance, so instead of using only the final block we learn, as in Peters et al. (2018), a convex combination of the input embedding and the outputs of all eight blocks,

$$z_i = \sum_{\ell=0}^{8} \mathrm{softmax}(\gamma)_\ell \, x_i^{(\ell)} \in \mathbb{R}^d, \tag{3}$$

where $x_i^{(\ell)}$ is observation i after block ℓ, with $\ell = 0$ the input embedding, and $\gamma \in \mathbb{R}^9$ is learned. The pooled feature is used per observation to build the correlogram channels of Section 3.4, and averaged over the valid observations of a series to supply the output-stage bias of Section 3.6.

3.4 LAG-DOMAIN CORRELOGRAM

The correlogram takes the place of the frequency search, mapping the observations to a fixed-length record of how strongly the series agrees with itself at each lag. No single lag determines the period, and the information lies in the pattern across lags, which Section 3.5 maps onto the period grid.

Pairs separated by a given lag are compared through a vector per observation,

$$u_i = \begin{bmatrix} m_i \\ W z_i \end{bmatrix} \in \mathbb{R}^C, \qquad W \in \mathbb{R}^{(C-1)\times d}, \qquad C = 24, \tag{4}$$

whose first entry is the normalised value and whose remaining entries are a learned linear map of the pooled feature z_i of Section 3.3. Keeping C small is what bounds the cost of the lag axis. Each channel is centred over the valid observations of its series, and $\sigma^2 = \frac{1}{n}\sum_i \|u_i\|^2$ is the total variance across channels, which normalises everything below.

The multiples of a period are not equally informative. For a strictly periodic series, observations separated by τ differ in phase by the fractional part of τ/P, so the same structure recurs at kP for $k = 1, 2, 3, \ldots$, with an absolute width set by P and not by k. A small period error displaces the k-th multiple in proportion to k, so distant multiples constrain the period more tightly. On a logarithmic lag axis that structure occupies an interval narrowing as $1/k$, in the same coordinate as the period label, so its sharpness transfers directly to the estimate. How far this remains useful depends on coherence and on the number of sampled pairs at long lags.

The logarithmic axis allocates resolution by relative precision rather than by absolute lag. A linear grid matching its resolution at the shortest lags would need about 6×10^8 bins to reach the end of the lag axis. We use a lag bin width of $w = 0.002$ in $\log_{10}$ lag, a relative width of about 0.46%, which is one quarter of the period bin width $W = 0.008$ in $\log_{10}$ period (about 1.9%), extended far enough past the period range to cover the multiples used in Section 3.5, which gives $n_{\mathrm{lag}} = 3224$. Because W is an integer multiple of w, each period bin lands exactly on a lag bin, which lets Section 3.5 read the multiples at fixed index offsets.

For a lag bin $b = [\tau_b, \tau_{b+1})$, let $b(i)$ be the observations j whose separation $t_j - t_i$ falls in b. Within a series we accumulate

$$S_{b,c} = \sum_i \sum_{j \in b(i)} u_{i,c}\, u_{j,c}, \qquad N_b = \sum_i |b(i)|, \qquad D_b = \sum_i \sum_{j \in b(i)} \|u_i - u_j\|^2, \tag{5}$$

and from these form the curves the model reads,

$$\rho_c(\tau_b) = \frac{S_{b,c}}{(N_b + \kappa)\,\sigma^2}, \qquad \delta(\tau_b) = \frac{D_b + 2\kappa\sigma^2}{2(N_b + \kappa)\,\sigma^2}, \qquad \eta(\tau_b) = \frac{\log(1 + N_b)}{\log(1 + n)}. \tag{6}$$

There is one ρ_c per channel, together with their sum, so a series leaves this stage as $C + 3$ curves of length n_{lag}. Each ρ_c measures the self-similarity of channel c at that lag, while the dispersion δ

measures the separation of the full feature vectors. Irregular sampling can leave stretches of the lag axis empty, so the last curve η records how much pair support each lag has, separating empty bins from bins with genuinely weak self-similarity.

Expanding $\|u_i - u_j\|^2$ gives $\delta \approx 1 - \sum_c \rho_c$, the standard relation between the variogram and the autocovariance of a stationary process (Cressie, 1993). For the raw-value channel, ρ_0 is the series autocovariance at that lag, normalised by the total variance across channels. The representation therefore retains the classical second-order signal and adds learned channels alongside it, and with $C = 1$, the raw-value channel alone, it reduces to the classical autocorrelation, the case isolated in Table 3. Fold-based dispersion statistics (Stellingwerf, 1978) are related to δ, since folding at a trial period aggregates the pairs near its multiples and therefore behaves like a weighted combination of δ at those lags. Section 3.5 makes this correspondence explicit.

A bin holding three pairs can dip as deeply as one holding three hundred, so $\kappa = 4$ pseudo-pairs are added to every bin. Thin bins are pulled toward $\rho \to 0$ and $\delta \to 1$ and empty bins are exactly neutral, so little support produces a weak signal rather than a misleading one. Sorting by time makes each $b(i)$ a contiguous range, so prefix sums evaluate the statistics without enumerating pairs, and the result is exact within the lag bins, since every ordered pair contributes to its bin with no interpolation and no subsampling. The cost, $O(n\, n_{\mathrm{lag}}\, C)$, is linear in n at a fixed number of lag bins.

3.5 HARMONIC COMB

The correlogram is indexed by lag and the output by period, and the harmonic comb maps each period bin to the lag locations of its multiples.

With the lag bin width $w = 0.002$ and the period bin width $W = 4w$ of Section 3.4, both measured from a common origin, bin p represents the period P_p with $\log_{10} P_p = \lambda_0 + pW$. The k-th multiple of P_p sits at

$$\log_{10}(kP_p) \;=\; \lambda_0 + pW + \log_{10} k, \tag{7}$$

and dividing by w and rounding to the nearest bin gives the index

$$\mathrm{idx}(k, p) \;=\; 4p + o_k, \qquad o_k \;=\; \mathrm{round}\!\left(\frac{\log_{10} k}{w}\right), \tag{8}$$

where the stride 4 is the ratio W/w. Because o_k depends only on k, the same fixed offsets apply to every period bin. For each P_p we read the correlogram at P_p, $2P_p$, up to KP_p, giving $M \in \mathbb{R}^{(C+3)\times K\times N_P}$, where $M[c, k, p]$ is curve c at the k-th multiple of P_p.

The number of useful multiples is determined by the lag resolution. The relative width of the feature at the k-th multiple is about $1/k$, or $1/(k \ln 10)$ in $\log_{10}$ units. Once this falls below the lag bin width w, the multiple can no longer be resolved as a separate feature, so $k_{\max} \approx 1/(w \ln 10) \approx 218$ and we use $K = 218$, where a lag grid equal to the period grid would have allowed only $k_{\max} \approx 54$. The record sets a second limit, since a multiple lying beyond it, or beyond the window of L observations, has no pairs, so the full comb applies only to periods below about $T/218$ and for longer periods η marks the unreached multiples as empty.

For each period bin p, the slice $M[:, :, p]$ holds the correlogram features at its successive multiples, and the comb learns how to combine them across channels and multiples. One special case uses only the raw-value correlation ρ_0 and weights every multiple equally, giving $\sum_k \rho_0(kP_p)$. By the relation of Section 3.4, this is equivalent, up to constants that do not depend on the period, to aggregating the value-channel dispersion over the same multiples. Classical fold-based statistics (Stellingwerf, 1978) perform a similar aggregation over the multiples of a trial period, so the learned comb generalises that fixed aggregation by choosing which channels and multiples to emphasise.

A weighted sum over (c, k) treats each multiple independently, but whether the feature at one multiple is reliable depends on those next to it. We therefore apply a small convolutional network to each period bin's $(C + 3) \times K$ slice, with dilated convolutions (Yu & Koltun, 2016) along k that combine information from neighbouring multiples. We also add a learned embedding of k at the input, which lets the network distinguish low multiples from high ones, since they differ in sharpness and in the number of pairs supporting them. All kernels have unit length along the period axis and the normalisation acts within each (k, p) position, so no two period bins are mixed at this stage, and comparison across them is left to the decoder of Section 3.6. The channel and multiple axes are then

flattened together, and a 1×1 convolution along the period axis reduces them to the decoder input $F_{\mathrm{comb}} \in \mathbb{R}^{256 \times N_P}$.

3.6 DECODER

The comb summarises the correlogram across the multiples of each period, and that aggregation can lose the local shape of the correlogram around the fundamental. The decoder therefore receives, for each period bin, both the comb summary F_{comb} and the uncompressed correlogram at the fundamental, $M[:, 1, :] \in \mathbb{R}^{(C+3) \times N_P}$, in which neighbouring period bins correspond to neighbouring lag bins, so that convolution along the period axis can use that local structure. Concatenating the two gives $256 + 27 = 283$ channels over $N_P = 512$ period bins.

A one-dimensional convolutional network with dilations along the period axis maps this $283 \times N_P$ input to N_P logits, one per period bin. Its dilations, and two layers with taps at fixed period ratios from $1/5$ to 5, are set so that a period together with its half, double, third and triple falls within a single receptive field, where the choice among them is made.

An MLP maps the series-level pooled feature of Section 3.3 and $\log_{10} s$ to N_P values added to the logits. Its output layer is zero-initialised, and it is the only path by which the scale of a series reaches the output.

The result is a distribution over the N_P period bins, which form a coarse set of candidates, scored by the network rather than tested one at a time against the observations, and because they are evenly spaced in log-period their resolution is fixed by the period rather than by the record length.

3.7 OUTPUT DISTRIBUTION, LOSS AND DECODING

We treat y as a distributional regression target rather than a class label (Imani & White, 2018; Farebrother et al., 2024), placing a Gaussian with a standard deviation of two bins on y and integrating it over each bin to give a soft target t, against which the logits are trained by cross-entropy,

$$\mathcal{L}_{\mathrm{ce}} = -\sum_{p} t_p \, \log \mathrm{softmax}(z)_p. \tag{9}$$

The soft target preserves the position of the label within a bin and avoids the discontinuity a one-hot target creates at bin boundaries.

The soft target captures proximity in period, but it does not distinguish an error at $2P$ from any other error at the same distance. Since a series with period P also repeats at its integer multiples, we add a hinge over the ratios $R = \{1/2, 2, 1/3, 3\}$. Let $q(a)$ denote the log of the softmax mass within two period bins of a, and define

$$\mathcal{L}_{\mathrm{harm}} = \frac{1}{|R|} \sum_{r \in R} \max\bigl(0,\ \mu + q(y + \log_{10} r) - q(y)\bigr). \tag{10}$$

This encourages the mass around the labelled period to exceed the mass around each ratio in R by at least $\mu = 1$ nat, with ratios whose targets fall outside the grid masked, and the total loss is $\mathcal{L} = \mathcal{L}_{\mathrm{ce}} + 0.3\,\mathcal{L}_{\mathrm{harm}}$.

To reduce period imbalance, training samples series with probability proportional to $\mathrm{count}(\mathrm{bin}(y))^{-\alpha}$, with $\alpha = 0.5$, which is square-root sampling (Kang et al., 2020). Because this changes the training prior, we add $\tau \log \pi_{\mathrm{train}}$ to the logits at inference, a post-hoc logit adjustment (Menon et al., 2021), where π_{train} is the label distribution of the training split and τ is set on validation.

A series that repeats every P also repeats every $2P$, so the distribution can carry more than one mode and an expectation over its full support can fall between them. We therefore take the softmax centroid within eight bins of the argmax, selecting one mode while keeping sub-bin precision.

3.8 LOCAL REFINEMENT

The estimator separates two questions, which period a series has, settled by Section 3.7, and its exact value, which a second stage refines without revisiting that choice. They need different precision,

since a period known to a few tenths of a percent is already identified but cannot yet fold the record, where over thousands of cycles an error of that size accumulates into several cycles of phase drift.

The refinement is therefore local. Starting from the first-stage estimate it considers only a narrow band around it, dividing the record into equal chunks, folding each at the current estimate and summarising it by the first Fourier components of its folded profile. A slightly wrong estimate makes the phase of that profile advance linearly from chunk to chunk, and the rate of advance gives the correction. A trained network reads this phase advance and proposes corrections, a small beam of the resulting candidates is carried forward, and repeating the step a few times while narrowing the band sharpens the estimate, a procedure we call the learned contraction. A Gauss–Newton polish (Nocedal & Wright, 2006) then fits an eight-harmonic model of the folded profile to the whole record.

The band is too narrow to contain any other multiple, so this stage cannot correct a wrong choice of multiple and is a precision step rather than a period finder, and since a classical local refinement inside the same band reaches comparable precision we claim no novelty for it. Its architecture, training and precision at each step are given in Appendix C.

4 DATA

The synthetic corpus, made with the generator of Anonymous (2026) (Appendix I), holds 600,002 series built from 98,066 waveform templates in 13 families. Each series paints a template onto the sampling pattern and noise of a real donor series, at a primary period drawn from its family's range, and that period is the only label it carries. Splitting by template gives 479,747 training, 60,028 validation and 60,227 test series, so no waveform seen in training reappears in test. Donors are shared across the split, with 90.4% of those in test also used in train, so the synthetic results test generalisation to unseen waveforms but not to unseen sampling patterns. The median series has 934 observations over 2,059 time units, about 4,900 cycles, and periods run from 0.03 to 119.7 time units. Accuracy drops below about 300 observations (Appendix G).

The real series are single-band forced-photometry light curves (Masci et al., 2023) from a public time-domain survey (Bellm et al., 2019). Their catalogue periods, from the public SCoPe release (Healy et al., 2024b), were themselves found by a period search on the same observations with the nested ELS_ECE_EAOV procedure, which combines Lomb–Scargle, conditional-entropy, and analysis-of-variance searches (Healy et al., 2024a). Thus, agreement with the catalogue shows that two estimators agree, not that either is correct. We keep the 174,337 of 555,973 labelled series whose catalogue period lies in the label range and folds the observations with $\theta \leq 0.70$ at 15 phase bins (Section 5). This sets aside series in which no periodic signal stands out from the noise, and the real results describe those that remain. Of these, 17,129 are held out from the template pool, the donor pool and every population statistic used to build the synthetic corpus.

5 EXPERIMENTS

Synthetic results are reported on the template-disjoint test split and real results on the held-out set of Section 4. The model is trained only on synthetic series, and the checkpoint, τ and decoding rule, all fixed on synthetic validation data, are carried to real series without fine-tuning or recalibration. Three seeds vary the initialisation and the order in which series are drawn while sharing a single split, and $\pm$ denotes the standard deviation across them. Every comparison with a baseline is paired on the same series, with $[\mathrm{lo}, \mathrm{hi}]$ its 95% bootstrap interval (Efron & Tibshirani, 1993) over 2,000 resamples. Accuracy is measured on the first-stage estimate of Section 3.7, and fold quality on the full two-stage estimate of Section 3.8, to which the catalogue period is also submitted so that both are compared at their own fold optimum.

Baselines. The periodogram is the Lomb–Scargle implementation of `astropy` (Astropy Collaboration et al., 2022; VanderPlas, 2018) with per-observation uncertainties. It searches periods from 0.01 to 120 time units, slightly wider than the label range, at 5 samples per peak, and returns the period of its highest peak. Phase dispersion minimisation (PDM; Stellingwerf, 1978), with ten phase bins, and conditional entropy (CE; Graham et al., 2013b), with ten phase bins and five value bins, search the same range under the same rule, with frequency steps of $1/(5T)$ for the median

record length T of each set, which gives 1,093,448 trial frequencies per series on the held-out set and 1,026,744 on the synthetic test split. One learned baseline uses LagFormer's head and loss and differs from it only in its input, the raw-value autocorrelation alone ($C = 1$, same corpus, same schedule), which isolates the representation. Three published learned estimators are retrained on the same corpus (Appendix E). Two controls bound what the period prior can supply, the mode of each family's training period distribution given the true family, and LagFormer with values permuted among timestamps.

Metrics. Let $\hat{P}$ be the estimated period and P the reference period. We report Acc_{10} and Acc_{20}, the fractions with $|\hat{P}/P - 1|$ below 0.1 and 0.2, the harmonic-tolerant Acc^{h}_{10}, which also accepts $\hat{P}$ within 10% of $P/2$ or $2P$, $\mathrm{MAE}_{\log} = \mathrm{mean}\,|\log_{10}\hat{P} - \log_{10}P|$, and the fraction of gross errors with $|\log_{10}(\hat{P}/P)| > 0.5$. On synthetic series these are computed against the true period. On real series the same quantities are computed against the catalogue period and reported as agreement, Agr_{10} and Agr^{h}_{10}, since the catalogue period is itself an estimate.

Fold quality. On real series we also compute the phase-binned dispersion statistic θ (Stellingwerf, 1978) with 18 phase bins averaged over 80 shifts of the bin edges, lower meaning a tighter fold. Of the bin counts from 10 to 30, 18 most often ranks two estimates of a synthetic series by their distance from the true period. We keep series whose catalogue period folds with $\theta < 0.7$ at 18 bins and compare two periods only when they agree on the harmonic, which throughout means to within 10%. Since θ can favour a harmonic, such as half the period for two dips per cycle, this measures fold quality at a fixed harmonic and does not necessarily identify the correct one.

6 Results

Table 1: First-stage estimate. Synthetic columns give accuracy on the test split ($n = 60{,}227$) against the true period. Real columns give agreement with the catalogue period on the held-out set ($n = 17{,}129$), which is not a measure of accuracy since that period was itself found by a search (Section 4). LagFormer is the mean $\pm$ standard deviation over three seeds, and bold marks the best estimator in each synthetic column. Retrained on our corpus (Appendix E): [c]Claytor et al. (2022), [d]Schochet et al. (2025), [e]Donoso-Oliva et al. (2023), [f]with FAN layers (Dong et al., 2025).

	Synthetic (accuracy)			Real (agreement)		
	Acc_{10}	Acc^{h}_{10}	$\mathrm{MAE}_{\log}$	Agr_{10}	Agr^{h}_{10}	$\mathrm{MAE}_{\log}$
Periodogram	69.27	94.25	0.1093	72.28	96.44	0.0906
PDM	85.04	**95.22**	0.0617	82.75	97.14	0.0572
CE	82.16	93.34	0.0834	80.51	96.31	0.0671
ACF-input network ($C = 1$)	89.51	94.69	0.0379	81.00	95.68	0.0676
Windowed-periodogram CNN[c,d]	25.72	34.86	0.1945	20.82	30.73	0.1806
Irregular-sampling encoder[e]	16.43	24.87	0.3981	17.98	29.49	0.3022
FAN encoder[f]	17.07	25.59	0.3792	19.25	30.25	0.2791
Prior only	22.52	30.06	0.2466	22.21	29.17	0.2641
LagFormer, shuffled times	6.30	14.56	0.5971	9.19	18.09	0.4583
LagFormer	**90.17**$\pm$0.08	94.65$\pm$0.07	**0.0360**$\pm$0.0002	80.64$\pm$0.12	95.57$\pm$0.09	0.0690$\pm$0.0004

Accuracy on synthetic series. Here the period is known independently, so Table 1 measures accuracy directly. LagFormer is ahead of every classical search on Acc_{10} and $\mathrm{MAE}_{\log}$, with paired intervals well clear of zero, and ahead of the periodogram by $+20.9$ points $[+20.6, +21.3]$. It also far outperforms the published learned estimators retrained on the same corpus. One of them uses LagFormer's own encoder, trained identically but with an MLP in place of the correlogram, comb and decoder (Appendix E), and reaches only 16.43%, so the gain comes from the lag-domain readout, not the encoder. Under harmonic-tolerant scoring, the margin closes and PDM is ahead, indicating that PDM is particularly prone to picking a harmonic over the fundamental. The controls rule out the period prior as the source of these results. The family prior is far behind every classical search and LagFormer on both sets, and permuting values among timestamps, which keeps the sampling and the values but breaks any periodic relation between them, drops the model below even the prior. Nearly half of the remaining errors are at $P/2$ or $2P$; Appendix D gives Acc_{20} and gross errors.

Agreement on real series. LagFormer agrees with the catalogue period more often than the periodogram does, by $+8.4$ points $[+7.6, +9.1]$, although under harmonic-tolerant scoring the periodogram is slightly ahead, since most of its disagreements are at half the catalogue period. It is level with CE ($\Delta_{\mathrm{CE}} = +0.13$ $[-0.44, +0.73]$) and behind PDM ($\Delta_{\mathrm{PDM}} = -2.11$ $[-2.68, -1.54]$). PDM's phase-binned variance statistic is closely related to the analysis-of-variance component of the catalogue search (Schwarzenberg-Czerny, 1989), so part of its lead may reflect a shared method. As the catalogue period is itself a search result, agreement measures consistency, not accuracy, so the estimates are also compared by fold quality below. What agreement does show is that LagFormer, trained only on synthetic series, comes close to the catalogue search's period without running a search, with an estimate that is differentiable and can be trained against.

Table 2: Share of held-out series on which each period, after the second stage, folds more tightly than the catalogue period (seed 42, $n = 12{,}827$). The subset keeps series on which all three seeds and the catalogue agree on the harmonic and the catalogue period folds with $\theta < 0.7$. The periodogram row uses the $12{,}359$ series that pass the same θ condition and share the catalogue harmonic.

	% tighter
LagFormer, first stage only	0.1
LagFormer, full	52.8 [52.0, 53.6]
Periodogram, full	49.4 [48.5, 50.2]
Catalogue, second stage (ceiling)	53.7 [52.8, 54.6]

Fold quality. Table 2 counts how often each period, after the second stage, folds more tightly than the catalogue period. The full LagFormer estimate does so on 52.6–52.9% of the common subset across the three seeds, nearly matching the 53.7% obtained by refining the catalogue period itself, while the refined periodogram is at 49.4%. The first stage alone does so on only 0.1%, showing that the improvement comes from the precision of the second stage, which reaches a median $|\Delta \log_{10} P|$ of 1.2×10^{-6} on synthetic series with 8 trial frequencies per series, against 6,147 for a classical local zoom (Appendix C).

Table 3: Ablations of selected components, first-stage estimate, seed 42, with τ tuned per run. Δ is the paired difference in synthetic Acc_{10} from the full model with its 95% bootstrap interval; the last column is held-out Agr_{10}. †: also seeds 43, 44 (pooled seed noise 0.19 synthetic / 0.37 real).

	Synthetic			Real
	Acc_{10}	$\mathrm{MAE}_{\log}$	Δ	Agr_{10}
Full model ($K = 218$, $C = 24$)†	90.20	0.0358	–	80.64
No per-candidate CNN†	88.42	0.0427	-1.78 $[-1.98, -1.58]$	80.69
Raw values only ($C = 1$)†	89.51	0.0379	-0.69 $[-0.86, -0.52]$	81.00

Ablations. Table 3 removes one component at a time. Both removals hurt synthetic accuracy, and both concern how each lag is read. Removing the per-candidate CNN leaves each lag a scalar, and raw values alone ($C = 1$) give the ACF-input network of Table 1. The encoder sharpens the estimate rather than changing its multiple: on series both get within 10%, its median error is 7% $[6, 8]$ lower on synthetic and 9% $[7, 10]$ lower against the catalogue on real series (Appendix G).

7 Conclusion

LagFormer regresses a continuous period from irregularly sampled series, reading self-similarity on a logarithmic lag axis where fixed offsets gather each candidate's multiples. Binning observation pairs by time difference, it needs no interpolation or regular cadence. Trained only on synthetic series, its first stage is more accurate on synthetic data than searches over 10^6 trial frequencies at 2.7–16$\times$ lower median CPU cost, and on real light curves its refined period folds nearly as tightly as the refined catalogue period. Recovering a period to a few parts per million from series typically observed less than once per cycle, with no fine global search, is the central result. Periods drawn per family let the model learn family period priors, and independently drawn periods break shape–period relations that only real data tests. The model also assumes periodic input, weakens below 300 observations and reads fewer multiples above $T/218$.

AI USE STATEMENT

We used generative AI tools during the preparation of this work to assist with research discussion, methodological critique, experimental planning, and manuscript drafting and editing. AI-assisted suggestions were reviewed and verified by the authors, and all experimental results, analyses, mathematical formulations, and scientific claims reported in this paper were independently checked by the authors. The authors take full responsibility for the final content of the paper.

ETHICS STATEMENT

This work uses astronomical photometry from a time-domain survey and a public catalogue of variable-star periods (Bellm et al., 2019; Masci et al., 2023; Healy et al., 2024b), together with synthetic series built from them. It involves no human participants and no personal or sensitive data, and we are not aware of a harmful use of a period estimator for stellar light curves. The generator of the synthetic corpus is described in a separate companion paper under review (Anonymous, 2026); this paper contributes the estimator and its evaluation and uses that generator only to produce training data. We have read and adhere to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

Section 3 describes the model, Section 4 describes the training and test data, and Section 5 gives the metrics, baselines and evaluation sets. Appendix A lists every hyperparameter and the training schedule, and Appendix I describes how the synthetic corpus is built. Appendices B and C give the exact computation of the correlogram and of the second stage, and Appendix E explains how each learned baseline was retrained. The catalogue periods are public (Healy et al., 2024b), and the light curves are forced photometry of survey images (Bellm et al., 2019; Masci et al., 2023). The supplementary material contains anonymous source code for both stages, the configuration of every reported run, the list of held-out series and the script that produces the fold-quality table (Table 2). Every result states the seeds it uses.

REFERENCES

Anonymous. Synthetic training sets no easier than the survey, 2026. Under review.

Astropy Collaboration, Adrian M. Price-Whelan, Pey Lian Lim, Nicholas Earl, Nathaniel Starkman, Larry Bradley, David L. Shupe, Aarya A. Patil, Lia Corrales, C. E. Brasseur, et al. The Astropy project: Sustaining and growing a community-oriented open-source project and the latest major release (v5.0) of the core package. *The Astrophysical Journal*, 935(2):167, 2022.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Albert E. Beaton and John W. Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, 16(2):147–185, 1974.

Eric C. Bellm, Shrinivas R. Kulkarni, Matthew J. Graham, Richard Dekany, Roger M. Smith, Reed Riddle, Frank J. Masci, George Helou, Thomas A. Prince, Scott M. Adams, et al. The Zwicky Transient Facility: System overview, performance, and first results. *Publications of the Astronomical Society of the Pacific*, 131(995):018002, 2019.

Kirsten Blancato, Melissa K. Ness, Daniel Huber, Yuxi (Lucy) Lu, and Ruth Angus. Data-driven derivation of stellar properties from photometric time series data using convolutional neural networks. *The Astrophysical Journal*, 933(2):241, 2022.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8:6085, 2018.

Zachary R. Claytor, Jennifer L. van Saders, Joe Llama, Peter Sadowski, Brandon Quach, and Ellis A. Avallone. Recovery of TESS stellar rotation periods using deep learning. *The Astrophysical Journal*, 927(2):219, 2022.

Noel A. C. Cressie. *Statistics for Spatial Data*. Wiley, revised edition, 1993.

Yihong Dong, Ge Li, Yongding Tao, Xue Jiang, Kechi Zhang, Jia Li, Jinliang Deng, Jing Su, Jun Zhang, and Jingjing Xu. FAN: Fourier analysis networks. In *Advances in Neural Information Processing Systems*, volume 38, pp. 73845–73874, 2025.

C. Donoso-Oliva, I. Becker, P. Protopapas, G. Cabrera-Vives, M. Vishnu, and H. Vardhan. ASTROMER: A transformer-based embedding for the representation of light curves. *Astronomy & Astrophysics*, 670:A54, 2023.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Counting out time: Class agnostic video repetition counting in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10384–10393, 2020.

R. A. Edelson and J. H. Krolik. The discrete correlation function: A new method for analyzing unevenly sampled variability data. *The Astrophysical Journal*, 333:646–659, 1988.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

Pablo A. Estévez, Pablo Huijse, Pablo Zegers, José C. Príncipe, and Pavlos Protopapas. Period detection in light curves from astronomical objects using correntropy. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2010.

Jesse Farebrother, Jordi Orbay, Quan Vuong, Adrien Ali Taiga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, Aviral Kumar, and Rishabh Agarwal. Stop regressing: Training value functions via classification for scalable deep RL. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 13049–13071. PMLR, 2024.

M. J. Graham, A. J. Drake, S. G. Djorgovski, A. A. Mahabal, C. Donalek, V. Duan, and A. Maker. A comparison of period finding algorithms. *Monthly Notices of the Royal Astronomical Society*, 434(4):3423–3444, 2013a.

Matthew J. Graham, Andrew J. Drake, S. G. Djorgovski, Ashish A. Mahabal, and Ciro Donalek. Using conditional entropy to identify periodicity. *Monthly Notices of the Royal Astronomical Society*, 434(3):2629–2635, 2013b.

Brian F. Healy, Michael W. Coughlin, Ashish A. Mahabal, Theophile Jegou du Laz, Andrew Drake, Matthew J. Graham, Lynne A. Hillenbrand, Jan van Roestel, Paula Szkody, LeighAnna Zielske, et al. The ZTF source classification project. III. A catalog of variable sources. *The Astrophysical Journal Supplement Series*, 272(1):14, 2024a.

Brian F. Healy, Michael W. Coughlin, et al. The data for "The ZTF Source Classification Project: III. A Catalog of Variable Sources". Zenodo, `https://doi.org/10.5281/zenodo.8410825`, 2024b.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Max Horn, Michael Moor, Christian Bock, Bastian Rieck, and Karsten Borgwardt. Set functions for time series. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 4353–4363. PMLR, 2020.

Huazhang Hu, Sixun Dong, Yiqun Zhao, Dongze Lian, Zhengxin Li, and Shenghua Gao. TransRAC: Encoding multi-scale temporal correlation with transformers for repetitive action counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18991–19000, 2022.

Ehsan Imani and Martha White. Improving regression performance with distributional losses. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 2157–2166. PMLR, 2018.

Ilay Kamai and Hagai B. Perets. Accurate and robust stellar rotation periods catalog for 82771 Kepler stars using deep learning. *The Astronomical Journal*, 169(2):59, 2025.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

N. R. Lomb. Least-squares frequency analysis of unequally spaced data. *Astrophysics and Space Science*, 39:447–462, 1976.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Ashish Mahabal, Kshiteej Sheth, Fabian Gieseke, Akshay Pai, S. George Djorgovski, Andrew Drake, and Matthew Graham. Deep-learnt classification of light curves. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8. IEEE, 2017.

Frank J. Masci, Russ R. Laher, Ben Rusholme, et al. A new forced photometry service for the Zwicky Transient Facility. arXiv:2305.16279, 2023.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. In *International Conference on Learning Representations*, 2018.

Brett Naul, Joshua S. Bloom, Fernando Pérez, and Stéfan van der Walt. A recurrent neural network for classification of unevenly sampled variable stars. *Nature Astronomy*, 2(2):151–155, 2018.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, second edition, 2006.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pp. 1310–1318. PMLR, 2013.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2227–2237, 2018.

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

K. Rehfeld, N. Marwan, J. Heitzig, and J. Kurths. Comparison of correlation analysis techniques for irregularly sampled time series. *Nonlinear Processes in Geophysics*, 18(3):389–404, 2011.

Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems*, volume 32, pp. 5320–5330, 2019.

Jeffrey D. Scargle. Studies in astronomical time series analysis. II. Statistical aspects of spectral analysis of unevenly spaced data. *The Astrophysical Journal*, 263:835–853, 1982.

Meir E. Schochet, Penelope Planet, Zachary R. Claytor, Jamie Tayar, and Adina D. Feinstein. 200,000+ deep learning–inferred periods of stellar variability from the All-Sky Automated Survey for Supernovae. *The Astrophysical Journal Supplement Series*, 282(1):10, 2025.

A. Schwarzenberg-Czerny. On the advantage of using analysis of variance for period search. *Monthly Notices of the Royal Astronomical Society*, 241(2):153–165, 1989.

David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, 1992.

Satya Narayan Shukla and Benjamin M. Marlin. Multi-time attention networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2021.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

R. F. Stellingwerf. Period determination using phase dispersion minimization. *The Astrophysical Journal*, 224:953–958, 1978.

Jacob T. VanderPlas. Understanding the Lomb–Scargle periodogram. *The Astrophysical Journal Supplement Series*, 236(1):16, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430, 2021.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 10524–10533. PMLR, 2020.

Xiao-Hui Xu, Qing-Feng Zhu, Xu-Zhi Li, Bin Li, Hang Zheng, Jin-Sheng Qiu, and Hai-Bin Zhao. A new period determination method for periodic variable stars. *Publications of the Astronomical Society of the Pacific*, 134(1041):114507, 2022.

Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations*, 2016.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 27268–27286. PMLR, 2022.

A Implementation and Training Details

This section lists the settings of the first stage (Section 3), how τ and the decoding rule were chosen, and how the cost of every estimator was measured.

Training and model selection. The three seeds share the configuration of Table 4–7 and differ only in the seed, which sets the initialisation and the order in which series are drawn. Validation Acc_{10} is computed on the 60,028 validation series every second epoch, and the checkpoint with the highest value is kept, which is epoch 6 for seeds 42 and 43 and epoch 8 for seed 44. Training took 31,512 s for seed 42 and 30,340 s for seed 43 on one RTX 6000 Ada, and 42,722 s for seed 44 on one RTX A6000.

Table 4: First-stage settings (1/4): input and encoder. Identical for the three seeds; the main text rounds W and w to 0.008 and 0.002.

Observations per series	$L = 1536$; contiguous window, random offset in training, start of record at evaluation
Time encoding	sinusoids at $d/2 = 192$ log-spaced periods from 0.002 to 1000 time units
Encoder	8 post-norm blocks (Vaswani et al., 2017), $d = 384$, 8 heads, feed-forward 2048, dropout 0.1 (Srivastava et al., 2014)
Observation dropout	8% of valid observations removed per series, training only
Layer pooling	softmax weights over the embedding and the 8 block outputs ($\gamma \in \mathbb{R}^9$), as in Peters et al. (2018)

Table 5: First-stage settings (2/4): correlogram and comb.

Channels	$C = 24$ (raw value and 23 learned)
Period grid	$N_P = 512$ bins over $\log_{10} P \in [-2,\ 2.08]$, $W = 4.08/512 = 0.00797$
Lag grid	$w = W/4 = 0.00199$, lags from 0.01 to 2.65×10^4 time units, $n_{\mathrm{lag}} = 3224$
Shrinkage	$\kappa = 4$ pseudo-pairs per lag bin
Comb	$K = 218$ multiples; offsets $o_k = \mathrm{round}(\log_{10} k/w)$, largest $o_{218} = 1174$; the $C + 3$ curves are normalised per series (one mean and variance) before the gather
Per-candidate CNN	1×1 stem to width 32, learned embedding of k, 3 residual blocks of kernel 3 along k with dilations 1, 2, 4 (Yu & Koltun, 2016), channel LayerNorm (Ba et al., 2016), GELU (Hendrycks & Gimpel, 2016)
Comb output	1×1 convolution to $F_{\mathrm{comb}} \in \mathbb{R}^{256 \times N_P}$

Table 6: First-stage settings (3/4): decoder and output.

Decoder input	$256 + 27 = 283$ channels, GroupNorm (Wu & He, 2018), kernel-5 stem to width 256
Harmonic taps	depthwise convolution with width-3 taps at ratios $\{\frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, 1, \frac{4}{3}, \frac{3}{2}, 2, \frac{5}{2}, 3, 4, 5\}$, before and after the dilated stack
Dilated stack	6 residual layers, kernel 5, dilations $1, 2, \ldots, 32$, GroupNorm(8), GELU, dropout 0.1
Series-level bias	MLP of the pooled feature and $\log_{10} s$, hidden 256, zero-initialised output
Soft target	Gaussian of standard deviation 2 bins integrated over each bin (Imani & White, 2018)
Harmonic hinge	weight 0.3, margin $\mu = 1$ nat, $R = \{\frac{1}{2}, 2, \frac{1}{3}, 3\}$, linear ramp over the first 15% of steps
Decoding	softmax centroid within 8 bins of the argmax

Table 7: First-stage settings (4/4): optimisation.

Optimiser	AdamW (Loshchilov & Hutter, 2019), weight decay 0.05, gradient norm clipped at 1.0 (Pascanu et al., 2013)
Learning rate	peak 7×10^{-4} (3.5×10^{-4} for encoder parameters), linear warm-up over the first 5% of steps, cosine decay to 0 (Loshchilov & Hutter, 2017)
Batch, epochs	64 series, 8 epochs over 479,747 training series, random initialisation
Balanced sampling	weight $\propto \text{count}^{-0.5}$ over 64 equal-width bins of $\log_{10} P$ (Kang et al., 2020)
Weight averaging	exponential moving average, decay 0.999; averaged weights are evaluated
Precision	automatic mixed precision (Micikevicius et al., 2018); correlogram accumulated in fp32 with fp64 times
Parameters	21,950,989 (encoder 17,346,688; correlogram, comb, decoder and bias 4,604,301)

Choice of τ and decoding. After training, we choose both the prior-adjustment strength τ and the decoding rule on the validation set. We evaluate $\tau \in \{0, 0.25, 0.5, 0.75, 1\}$ with two decoding rules: the centroid of the bins around the highest-probability bin, and the mean over the full predicted distribution. For seeds 42, 43, and 44, the best τ values are 0.5, 0.75, and 0.25, respectively. Across the three validation sets, these centroid choices give 94.42% Acc_{10}, compared with 90.61% for the full-support mean. We therefore use centroid decoding, with τ selected separately on validation for each seed.

Cost measurement. Table 8 reports processor time per series for each estimator and for each stage of LagFormer. We draw 96 series at random (seed 0) from the real held-out set and pass them one at a time through the code used to produce the reported predictions, moved to the CPU. The measurements use 16 concurrent worker processes, each limited to one thread, on a 32-core AMD Ryzen Threadripper 9970X; one warm-up series per process is excluded. The periodogram, PDM and CE search the grid of Section 5, which contains 1,093,448 trial frequencies for this set, with one set of phase bins as in Table 1. On the GPU, the first stage processes all 174,337 real series in 516–522 s per seed at batch size 128, including data loading, while the learned contraction refines the 60,227 synthetic test series in 73–76 s per estimator at batch size 48.

On one CPU thread, the LagFormer first stage takes 1,339 ms per series at the median, compared with 3,586 ms for the periodogram, 10.8 s for PDM, and 21.0 s for CE. The learned contraction takes 80 ms and the Gauss–Newton polish 6.4 ms at the median, so the full two-stage estimate takes about 1.43 s, still $2.5\times$ faster than the periodogram; the classical local periodogram zoom takes 297.3 ms. Thus the LagFormer first stage is $2.7\times$ faster than the median periodogram search and $8\times$ and $16\times$ faster than PDM and CE, respectively, although it is slower than the fastest 10% of periodogram runs. PDM and CE show substantially wider timing variation with the number of observations. On the CPU, the first stage reproduces the stored GPU estimates to within 8×10^{-4} in $\log_{10} P$ on all 96 series; the full timing distributions are given in Table 8.

Table 8: Processor time per series on one CPU thread, in milliseconds, over 96 real held-out series (median and 10th and 90th percentiles). Classical searches above the line, the stages of LagFormer and the classical local refinement below it. The second-stage rows take the seed-42 first-stage estimate as input.

	median	10th pct.	90th pct.
Periodogram (fast, $\sim 10^6$ trials)	3,586	776	4,141
PDM (10 phase bins)	10,811	4,258	34,185
CE (10×5 bins)	21,020	3,800	35,480
LagFormer first stage	1,339	1,232	1,500
Learned contraction (8 rungs)	79.9	76.2	83.0
Gauss–Newton polish (30 steps)	6.4	4.8	8.2
Local periodogram zoom ($3 \times 2{,}049$)	297.3	203.0	411.7

B CORRELOGRAM COMPUTATION

This section gives the algorithm that evaluates Eq. (5) of Section 3.4 exactly without enumerating pairs, its cost, and its measured runtime.

Sorting the observations by time makes the partners of observation i in lag bin $b = [\tau_b, \tau_{b+1})$ a contiguous index range $[r_{i,b},\ r_{i,b+1})$, where $r_{i,b}$ is the first index j with $t_j \geq t_i + \tau_b$. Adjacent bins share an endpoint, so each observation needs $n_{\mathrm{lag}} + 1$ binary searches. With prefix sums of the channel vectors, of their squared norms and of the valid-observation indicator, every sum over a partner range is a difference of two prefix entries, and Algorithm 1 accumulates $S_{b,c}$, N_b and D_b over all ordered pairs with $t_j - t_i \in [\tau_b, \tau_{b+1})$. Each such pair enters its bin once, so the result equals direct enumeration; padded slots are placed after every valid observation and have no partners. The cost is $O(n\, n_{\mathrm{lag}} \log n)$ for the searches and $O(n\, n_{\mathrm{lag}}\, C)$ for the accumulation, and memory is bounded by processing the lag bins in chunks. Training uses a fused GPU kernel that evaluates the same sums.

Algorithm 1 Exact log-lag correlogram by prefix sums (one series).

Require: times $t_{1..n}$ sorted, centred channel vectors $u_{1..n} \in \mathbb{R}^C$, lag edges $\tau_{0..n_{\mathrm{lag}}}$

$Z_0 \leftarrow 0,\ Q_0 \leftarrow 0,\ V_0 \leftarrow 0$; **for** $j = 1..n$: $Z_j \leftarrow Z_{j-1} + u_j, Q_j \leftarrow Q_{j-1} + \|u_j\|^2, V_j \leftarrow V_{j-1} + 1$
for $i = 1..n$ **and** $b = 0..n_{\mathrm{lag}}$ **do**
  $r_{i,b} \leftarrow \min\{j : t_j \geq t_i + \tau_b\}$ ▷ binary search
end for
for $b = 0..n_{\mathrm{lag}} - 1$ **do**
  $S_{b,:} \leftarrow \sum_i u_i \odot (Z_{r_{i,b+1}} - Z_{r_{i,b}})$; $N_b \leftarrow \sum_i (V_{r_{i,b+1}} - V_{r_{i,b}})$
  $D_b \leftarrow \sum_i \left[\|u_i\|^2 (V_{r_{i,b+1}} - V_{r_{i,b}}) + (Q_{r_{i,b+1}} - Q_{r_{i,b}})\right] - 2 \sum_c S_{b,c}$
end for
return S, N, D, from which ρ_c, δ and η follow by Eq. (6)

Table 9 compares Algorithm 1 with direct enumeration of all ordered pairs on one CPU thread. The two implementations agree to within 2.8×10^{-6} relative error on S and D across the series used in this paper. Direct enumeration is faster at $n = 1536$ (185 ms versus 260 ms), because the number of pairs, about $n^2/2$, is still smaller than $n\, n_{\mathrm{lag}}$. As n grows, the prefix-sum implementation becomes faster: at $n = 3072$ it takes 591 ms versus 716 ms, and at $n = 12{,}288$ it takes 3.65 s versus 11.1 s.

Table 9: Correlogram time on one CPU thread for the production grid ($C = 24$, $n_{\mathrm{lag}} = 3224$), median of 5 runs, for n observations at uniformly random times over 2,059 time units. "Pairs" counts ordered pairs whose lag falls on the grid. Both implementations give the same statistics to the relative difference stated in the text.

n	pairs	prefix sums (ms)	enumeration (ms)
128	8,128	25	2
256	32,639	47	6
512	130,814	90	22
768	294,522	131	44
1,024	523,767	173	85
1,536	1,178,868	260	185
3,072	4,717,009	591	716
6,144	18,871,097	1,690	2,790
12,288	75,490,619	3,654	11,148

C LOCAL REFINEMENT

This section specifies the second stage of Section 3.8: the observable, the correction network and its training, the contraction schedule, the Gauss–Newton polish, the precision after each step, a classical refinement in the same band, and the accuracy of the full estimate.

Observable. A hypothesis is a frequency f with a band half-width b believed to contain the true frequency. The record is divided into equal chunks of length $T_s = 0.15/b$, adjusted so that there are between 24 and 512 chunks. In chunk c, observations are folded at f into 32 phase bins with weights $1/e_i^2$, and the first two Fourier coefficients of the folded profile give $z_{c,h}$, $h \in \{1, 2\}$. If the true frequency is $f + \epsilon$, then $z_{c,h} \approx A_h \exp(2\pi i\, h\epsilon t_c)$, so the profile phase advances linearly with the chunk time t_c. The network reads

$$S_{d,h} = \sum_c z_{c+d,h}\, \overline{z_{c,h}}, \qquad d = 1, \ldots, 32, \tag{11}$$

whose phase is $2\pi h\, x\, \nu_d$, with $x = \epsilon/b \in [-1, 1]$ and $\nu_d = 0.15\, d$. The unknown profile phase cancels in the product, leaving a ramp whose slope depends only on the normalized frequency offset x. At the largest lag, $\nu_{32} = 4.8$, the phase can therefore wrap by up to 4.8 cycles for $h = 1$ and 9.6 cycles for $h = 2$.

Network. Each lag gives a token with 11 features: for each harmonic, the cosine, sine and angle of $S_{d,h}$, together with its log amplitude relative to the mean over lags; these are accompanied by ν_d, the log pair count and a validity flag. A sinusoidal encoding of ν_d at 8 frequencies is appended. A global token carries $\log_{10} P$, $\log_{10} T$, $\log_{10}(bT)$, the rung index, $\log_{10} n$, and the fraction of valid lags, so that the lag tokens themselves carry no period or record-length information. The sequence consists of this global token, 24 learned query tokens, and the 32 lag tokens, and passes through 6 pre-norm Transformer layers (Xiong et al., 2020) ($d = 256$, 8 heads, feed-forward width 1024, GELU, dropout 0.1). Each query output gives one mixture component with weight logit a_k, offset $x_k = \tanh(\cdot) \in [-1, 1]$, and width $\sigma_k = 3 \times 10^{-4} + \mathrm{softplus}(\cdot)$, expressed in units of b; the global output gives a logit p that the band contains the truth. The network has 5,018,396 parameters.

Contraction. Each mixture component proposes a child hypothesis $f_k = f + x_k b$ with half-width

$$b_k = \min\{b, \max(4\sigma_k b,\ 0.05/T)\}, \tag{12}$$

so the band can only contract and never becomes narrower than 0.05 cycles over the record. Children are ranked by their component score, $\log \mathrm{softmax}(a)_k + \log \mathrm{sigmoid}(p)$, added to the score of their parent, and a beam of one retains the highest-scoring child. The initial band is centered on P_0 with half-width $b_0 = 0.0125/P_0$, corresponding to $\pm 1.25\%$ in frequency and therefore excluding $P_0/2$ and $2P_0$. We repeat this contraction for eight rungs, each time evaluating the observable at the selected frequency; the final estimate is $1/f$. Thus the learned ladder uses only eight observable evaluations per series.

Training. The contraction network was trained once, before the first-stage model in this paper, on 59,520 training and 3,000 validation series from an earlier synthetic corpus generated with the same replay of real sampling patterns and noise. Each series starts from a period estimate produced by an earlier first-stage model; we retain only cases within 10% of the true period, ensuring that the truth lies inside the initial band. At each rung, for every hypothesis whose band contains the truth, we minimize the negative log-likelihood of

$$x_{\mathrm{true}} = \frac{f_{\mathrm{true}} - f}{b} \tag{13}$$

under the predicted Gaussian mixture, together with 0.3 times the binary cross-entropy for the band-containment probability p over all hypotheses. The loss at rung $r \in \{0, \ldots, 7\}$ is weighted by 0.8^{7-r}, giving later contractions greater weight, and gradients are detached between rungs. During training only, if the selected child leaves the true period outside its band while another proposed child still contains it, that child is retained for the next rung. We optimize with AdamW (Loshchilov & Hutter, 2019) (learning rate 3×10^{-4}, weight decay 0.01), batch size 96, 5% warm-up, gradient clipping at 1.0, and 25 epochs; epoch 16 is selected on validation. The network is not retrained on the corpus of Section 4.

Band width. The initial half-width is the only second-stage hyperparameter selected using first-stage outputs. We evaluated $b_0/P_0 \in \{0.005, 0.0075, 0.01, 0.0125, 0.015, 0.02\}$ on a random 80% of the synthetic test split (48,182 series), using the seed-42 first-stage estimates. We selected the value with the lowest median relative period error after refinement among the 43,464 series whose first-stage estimate was within 10% of the true period. The six choices give median errors from 5.89×10^{-6} to 6.67×10^{-6}, with 0.0125 giving the smallest error.

Gauss–Newton polish. For a fixed frequency ν, we fit a K-harmonic Fourier model with a constant term to all observations by weighted least squares, using weights $1/e_i^2$ normalized to mean one. At fixed ν, the Fourier amplitudes and the weighted residual sum $\chi^2(\nu)$ are available in closed form. Because the amplitudes minimize χ^2 at fixed ν, its derivative with respect to frequency, holding the fitted amplitudes fixed, is

$$\frac{d\chi^2}{d\nu} = -2\sum_i w_i r_i \, \partial_\nu \hat{m}_i. \tag{14}$$

We then take the Gauss–Newton step (Nocedal & Wright, 2006)

$$\Delta\nu = -\frac{d\chi^2/d\nu}{2\sum_i w_i (\partial_\nu \hat{m}_i)^2}, \qquad |\Delta\nu| \le 2 \times 10^{-3}\nu, \tag{15}$$

and accept the step only if χ^2 does not increase. We use $K = 8$ and 30 steps, starting from the ladder output. On the synthetic test split, the median $|\Delta \log_{10} P|$ after the polish for seed 42 is 1.66, 1.27, 1.20, 1.19, 1.21, 1.24 and 1.29×10^{-6} for $K = 2$, 4, 6, 8, 12, 16 and 20; the full step-wise results are given in Table 10. No trial frequencies are enumerated.

Classical local refinement as a control. To isolate the contribution of the learned contraction, we replace it with a local two-term periodogram (VanderPlas, 2018), using per-observation uncertainties, restricted to the same initial band around the first-stage estimate. The periodogram evaluates 2,049 frequencies across the band, then twice 2,049 frequencies across ± 4 steps of the previous maximum, and a parabolic fit to the final peak gives the local estimate (6,147 evaluations per series). On the synthetic test split, this classical refinement reaches a median $|\Delta \log_{10} P|$ of 1.57×10^{-6} (seed 42), below the 2.46×10^{-6} of the learned ladder, while the ladder is closer to the true period on 36.6% $[36.2, 37.0]$ of series. We then apply the same Gauss–Newton polish to both estimates, so the comparison isolates how the two methods locate the frequency within the initial band. After polishing, the medians are 1.11×10^{-6} for the classical refinement and 1.19×10^{-6} for the learned ladder, and the polished ladder is the closer of the two on 51.1% $[50.7, 51.5]$ of series. For seeds 43 and 44 the corresponding shares are 36.7% and 51.2%; the medians are in Table 10. The periodogram zoom moves 0.25% to 0.31% of these series, depending on the seed, to a different multiple. The learned ladder evaluates 8 frequencies per series, compared with 6,147 for the classical zoom.

Table 10: Precision of the second stage on the synthetic test split. Entries are the median $|\Delta \log_{10} P|$ after each contraction rung, evaluated on series whose ladder output is within 0.01 in $\log_{10} P$ of the true period (n per seed in the last row). The half-width column gives the median band passed to the next rung, and "truth in band" gives the fraction of these series for which the true frequency remains inside that band (both for seed 42). The classical local periodogram starts from the same first-stage estimate and initial band. All three seeds are shown; the Gauss–Newton rows give the final polished estimates.

	median $\lvert\Delta \log_{10} P\rvert$			half-width	truth in
	seed 42	seed 43	seed 44	(dex, s42)	band (s42, %)
First-stage estimate	7.10×10^{-4}	8.83×10^{-4}	5.73×10^{-4}	5.40×10^{-3}	99.0
after rung 1	2.73×10^{-5}	3.02×10^{-5}	2.45×10^{-5}	1.41×10^{-4}	96.8
after rung 2	3.05×10^{-6}	3.14×10^{-6}	3.02×10^{-6}	1.25×10^{-5}	94.0
after rung 3	2.50×10^{-6}	2.50×10^{-6}	2.49×10^{-6}	9.43×10^{-6}	92.9
after rung 4	2.46×10^{-6}	2.47×10^{-6}	2.45×10^{-6}	8.99×10^{-6}	92.6
after rung 5	2.46×10^{-6}	2.47×10^{-6}	2.45×10^{-6}	8.85×10^{-6}	92.5
after rung 6	2.46×10^{-6}	2.47×10^{-6}	2.45×10^{-6}	8.80×10^{-6}	92.5
after rung 7	2.46×10^{-6}	2.47×10^{-6}	2.45×10^{-6}	8.77×10^{-6}	92.4
after rung 8	2.46×10^{-6}	2.48×10^{-6}	2.46×10^{-6}	8.75×10^{-6}	92.4
+ Gauss–Newton polish	1.19×10^{-6}	1.19×10^{-6}	1.18×10^{-6}		
Local periodogram zoom	1.57×10^{-6}	1.57×10^{-6}	1.57×10^{-6}		
+ Gauss–Newton polish	1.11×10^{-6}	1.12×10^{-6}	1.11×10^{-6}		
n (same harmonic)	53,299	53,222	53,263		

What the ladder does. The learned contraction reduces the median error by more than an order of magnitude in the first three rungs, from 7.10×10^{-4} to 3.05×10^{-6} for seed 42, and then continues to tighten the band until the estimate stabilizes at approximately 2.46×10^{-6}. The probability that the true frequency remains inside the retained band falls from 99.0% initially to 92.4% after eight rungs, while the median band width contracts from 5.40×10^{-3} to 8.75×10^{-6}. Thus the ladder trades a small amount of coverage for a progressively narrower local search region.

Learned versus classical local refinement. The local periodogram reaches a comparable precision, with median errors of 1.57×10^{-6} before and 1.11×10^{-6} after the same Gauss–Newton polish, compared with 2.46×10^{-6} and 1.19×10^{-6} for the learned ladder. The distinction is therefore not that the learned contraction is a more precise local optimizer; rather, it reaches comparable precision with eight learned frequency evaluations instead of the 6,147 evaluations used by the classical zoom. Because the initial band excludes $P_0/2$ and $2P_0$, both procedures refine the selected harmonic rather than reconsidering the first-stage period choice.

D Error Breakdown

This section reports Acc_{20}, the gross-error rate, and the harmonic structure of first-stage errors, supporting Sections 1 and 6.

We count an estimate as an error when $\left|\hat{P}/P - 1\right| \geq 0.1$, and assign each error to ratio r when $\left|\hat{P}/(rP) - 1\right| < 0.1$. Table 11 shows the resulting error structure. On synthetic series, 80.7% of the periodogram's errors occur at half the true period, compared with 33.8% for LagFormer. On real series, the same pattern is even stronger for the searches: 86.9% of periodogram errors occur at half the catalogue period, compared with 31.6% for LagFormer. LagFormer's most common real-data error is instead twice the catalogue period, accounting for 45.5% of its errors. The raw-value correlogram network ($C = 1$) shows a similar harmonic structure, with errors concentrated at simple multiples on both datasets. Gross errors remain uncommon for all estimators; on real series, the periodogram and LagFormer have nearly identical rates of 1.44% and 1.45%.

Table 11: First-stage Acc_{20} (%), gross errors ($|\log_{10}(\hat{P}/P)| > 0.5$, % of all series), number of errors ($|\hat{P}/P - 1| \geq 0.1$), and share of errors (%) at each ratio $\hat{P}/P$ within 10%. Synthetic: test split, $n = 60{,}227$; real: held-out set, $n = 17{,}129$, against the catalogue period. LagFormer is the mean over three seeds and the $C = 1$ network is seed 42, as in Table 1.

	Synthetic					Real				
	Periodogram	PDM	CE	C=1	LagFormer	Periodogram	PDM	CE	C=1	LagFormer
Acc_{20}	69.45	85.23	82.44	90.09	90.86	72.57	83.00	80.79	81.71	81.41
Gross errors	2.28	2.00	3.35	1.07	1.10	1.44	1.07	1.47	1.35	1.45
Errors (n)	18,510	9,009	10,742	6,319	5,919	4,748	2,955	3,338	3,255	3,316
$\hat{P}/P = 1/2$	80.7	59.4	56.8	35.0	33.8	86.9	76.3	71.4	35.8	31.6
$\hat{P}/P = 2$	0.6	8.7	5.9	14.4	11.8	0.2	7.1	9.6	41.5	45.5
$\hat{P}/P = 1/3$	6.1	9.4	8.0	11.6	12.8	3.6	4.7	4.3	3.4	3.4
$\hat{P}/P = 3$	0.4	1.2	1.7	1.6	1.7	0.3	0.4	0.8	0.9	1.2
Other	12.2	21.3	27.6	37.4	40.0	8.9	11.5	13.8	18.5	18.3

E Retrained Learned Estimators

This section describes how the published learned estimators of Table 1 were retrained on our corpus. All use the template-disjoint split of Section 4, are selected on its validation part, and are scored on the same test and held-out series as LagFormer.

Windowed-periodogram CNN. The direct regressor of Claytor et al. (2022) and Schochet et al. (2025) reads an image of periodogram power against time. For each series we compute the weighted Lomb–Scargle power of the reference implementation at 128 time centres and 128 periods, each observation's weight tapered by a Gaussian window of width $2/\nu$ around the centre, average-pool the image to 64×64 and scale it to 8 bits per image. The convolutional network and its Laplace negative log-likelihood are used unchanged. Three changes follow from the task: the period axis spans 0.025 to 125 time units on a logarithmic scale instead of 1 to 30 on a linear one, the target is $(\log_{10} P + 2)/4.08$ so that it covers LagFormer's output range, and cells with fewer than two effective observations are set to zero instead of being interpolated. Training follows the published recipe: Adam (Kingma & Ba, 2015), the learning rate reduced by a factor 0.7 after 3 epochs without improvement in validation loss, batch 100, and early stopping after 20 such epochs. Four variants completed: channel widths $(8, 16, 32)$ and $(64, 128, 256)$ at the published initial learning rate 10^{-5}, and $(32, 64, 128)$ at 10^{-4} and 10^{-3}. We report the one with the lowest validation loss, widths $(32, 64, 128)$ at 10^{-4}, stopped at epoch 64 with its best weights from epoch 44. It reaches validation Acc_{10} of 26.06%, test Acc_{10} of 25.72% and held-out Agr_{10} of 20.82%. The variant at 10^{-3} has a higher validation Acc_{10} (27.19%) and a higher validation loss.

Irregular-sampling encoder. This row uses the encoder of Section 3.3, whose time encoding and block structure follow Donoso-Oliva et al. (2023), with the layer pooling of Section 3.3 and a three-layer MLP

($384 \to 256 \to 128 \to 512$) in place of the correlogram, comb and decoder. It is trained with LagFormer's output distribution, loss, sampling and schedule (Tables 4–7), and τ is chosen on validation in the same way. Its validation Acc_{10} is between 15.77% and 15.98% at every evaluation from epoch 2 to epoch 8, and its final validation loss, 5.33, is close to 5.53, the loss of predicting the training label distribution for every series. The row therefore measures an encoder with a pooled readout that learns little about the period under this schedule. It reaches test Acc_{10} of 16.43% and held-out Agr_{10} of 17.98%.

FAN encoder. This row is the same pooled-encoder architecture as the irregular-sampling encoder above, except that each feed-forward block replaces its two linear layers with gated Fourier Analysis Network (FAN) layers following Dong et al. (2025), using the authors' Transformer time-series setting in which both projections are gated FAN layers with dropout between them and no intermediate activation. The rest of the model, the training schedule and the choice of τ are unchanged, so the comparison isolates whether a periodicity-oriented FFN improves a direct readout from the encoder. Validation Acc_{10} rises only from 14.99% at epoch 2 to 16.21% at epoch 8, and the final validation loss is 5.27, again near the label-distribution floor of 5.53. On the reported splits it reaches test Acc_{10} of 17.07% and held-out Agr_{10} of 19.25%, slightly above the MLP feed-forward encoder and still far below LagFormer and every classical search. Substituting FAN for the MLP therefore does not overcome the pooled encoder's failure to recover the period from these sparse series.

F Fold Quality

This section gives Table 2 for each seed and compares the first-stage estimate with the full two-stage estimate, supporting the fold-quality paragraph of Section 6.

Fold quality is θ with 18 phase bins averaged over 80 shifts of the bin edges. A period folds more tightly than the catalogue period when its θ is lower than that of the catalogue period as published. The first three columns of Table 12 use the 12,827 series of Table 2, and the last two each estimator's own subset of the 17,097 held-out series whose catalogue period also folds with $\theta < 0.7$ at these 18 bins, out of the 17,129 held-out series of Section 4, which are selected with the same threshold at 15 bins, and on which the estimator agrees with the catalogue on the harmonic. The full estimate folds more tightly than the catalogue period on 52.6% to 52.9% of the common subset for the three seeds, the ladder alone on 18.4% to 18.7%, and the first stage alone on 0.1%. Polishing the catalogue period itself gives 53.7%.

Table 12: Share (%) of real held-out series on which a period, after the stages listed, folds more tightly than the catalogue period as published. Left: the 12,827 series on which all three seeds and the catalogue agree on the harmonic and the catalogue period folds with $\theta < 0.7$. Right: each estimator's own subset, sizes given.

	% tighter than catalogue (n = 12,827)			own same-harmonic subset	
	seed 42	seed 43	seed 44	n (s42/43/44)	% tighter (s42/43/44)
LagFormer, first stage only	0.1	0.1	0.1	–	–
LagFormer, first stage + ladder	18.7	18.4	18.7	13,792/13,809/13,767	18.7/18.4/18.7
LagFormer, full	52.8	52.6	52.9	13,792/13,809/13,767	52.1/52.0/52.3
Periodogram, full		–		12,359	49.4
Catalogue, second stage		53.7		17,097	52.5

The second stage runs on every series and moves the period only within the band around the first-stage estimate, so it cannot change which multiple was chosen. Table 13 confirms this on the synthetic test split: accuracy is unchanged and only $\mathrm{MAE}_{\log}$ falls.

Table 13: First-stage and full two-stage estimates on the synthetic test split (n = 60,227). LagFormer is the mean over three seeds. The periodogram row passes the periodogram's period through the same second stage.

	Acc_{10}	Acc_{20}	$\mathrm{MAE}_{\log}$
LagFormer, first stage only	90.17	90.86	0.0360
LagFormer, full	90.17	90.86	0.0353
Periodogram, first stage only	69.27	69.45	0.1093
Periodogram, full	69.26	69.45	0.1093

G ACCURACY BY PERIOD, SAMPLING AND FAMILY

This section breaks the first-stage results of Table 1 down by period, by the number of observations and by waveform family, to show where LagFormer and the periodogram each agree with the reference. Series with more than 1,536 observations are windowed to 1,536 (Section 3.2).

Period range. Table 14 shows accuracy across period ranges. On synthetic series, LagFormer remains between 88.6% and 91.7% Acc_{10} across all four ranges, while the periodogram ranges from 63.3% to 80.2%, showing that LagFormer's advantage is not confined to a particular period scale. On real series, the comparison depends strongly on period: below 0.2 time units, the periodogram agrees with the catalogue on 98.6% of series versus 85.4% for LagFormer. Above 0.2, LagFormer is ahead in every range, with margins of 26.3, 35.1, and 11.2 points for $0.2 \le P < 1$, $1 \le P < 5$, and $P \ge 5$, respectively.

Table 14: First-stage Acc_{10} (synthetic) and Agr_{10} (real), %, by period range (time units), for LagFormer (mean over three seeds) and the periodogram. Synthetic: test split; real: held-out set.

	Synthetic			Real		
	n	LagFormer	Periodogram	n	LagFormer	Periodogram
$P < 0.2$	13,329	88.6	78.1	8,005	85.4	98.6
$0.2 \le P < 1$	33,577	91.1	66.5	7,089	76.2	49.9
$1 \le P < 5$	10,119	88.8	63.3	1,662	78.8	43.7
$P \ge 5$	3,202	91.7	80.2	373	70.7	59.5

Number of observations. Table 15 shows the effect of sampling density. Both estimators generally improve as more observations are available. On synthetic series, LagFormer rises from 73.5% Acc_{10} for fewer than 300 observations to 92.9% for the 1,536-observation window, while the periodogram rises from 63.5% to 69.0%. The real series show the same overall trend, although LagFormer trails the periodogram in the sparsest bin: 48.4% versus 53.9% agreement among the 369 series with fewer than 300 observations. Above 300 observations, LagFormer is ahead in every bin, reaching 81.9% agreement for the 1,536-observation window.

Table 15: First-stage Acc_{10} (synthetic) and Agr_{10} (real), %, by number of observations n, for LagFormer (mean over three seeds) and the periodogram. Synthetic: test split; real: held-out set. Series with more than 1,536 observations are windowed to 1,536.

	Synthetic			Real		
	n	LagFormer	Periodogram	n	LagFormer	Periodogram
$n < 300$	3,690	73.5	63.5	369	48.4	53.9
$300 \le n < 600$	14,804	87.8	68.4	1,628	76.4	70.4
$600 \le n < 1000$	16,243	92.0	70.3	4,912	82.0	73.6
$1000 \le n < 1536$	15,461	92.8	70.5	6,572	81.8	72.8
$n = 1536$ (window)	10,029	92.9	69.0	3,648	81.9	72.2

Waveform family. Table 16 breaks the results down by source family. On real series, much of LagFormer's advantage comes from eclipsing systems, where the periodogram frequently returns half the catalogue period: LagFormer agrees on 83.8% of EA series versus 41.9% for the periodogram, and on 94.1% of EB series versus 18.9%. On the pulsating families RRAB, RRC, and DSCT, the two methods are much closer, differing by at most 3.4 points, while EW is the main family where the periodogram remains ahead, 84.2% versus 82.3%. The synthetic family results show a similar range of behavior: both methods perform very well on families such as CEPH and RRAB, while the periodogram has particularly low accuracy on EB and BIN_UNTYPED. Overall, the family breakdown shows that the real-data difference is concentrated in waveform classes where harmonic selection is especially consequential.

Table 16: First-stage Acc_{10} (synthetic) and Agr_{10} (real), %, by waveform family, for LagFormer (mean over three seeds) and the periodogram. Synthetic: test split; real: held-out set. Family labels are those of the source catalogue; "–" marks cells with fewer than 20 series.

	Synthetic			Real		
	n	LagFormer	Periodogram	n	LagFormer	Periodogram
ACCRETOR	0	–	–	50	15.3	24.0
BIN_UNTYPED	4,645	91.6	49.3	2,309	83.5	48.9
CEPH	4,734	98.9	96.0	33	87.9	87.9
DSCT	4,552	92.2	95.5	184	92.8	96.2
EA	4,593	91.7	39.2	332	83.8	41.9
EB	4,749	96.5	6.3	90	94.1	18.9
ECL_UNTYPED	4,581	85.9	50.9	1,822	77.5	53.0
EW	4,600	92.1	78.4	7,857	82.3	84.2
PNP_UNTYPED	4,672	79.6	66.4	3,436	75.4	71.8
PULS_UNTYPED	4,678	84.6	81.9	252	79.6	81.3
RRAB	4,566	98.8	98.0	240	98.5	97.9
RRC	4,631	97.1	98.1	122	93.2	93.4
RRLYR_UNRES	4,630	96.1	93.7	210	93.0	93.8
SIN	0	–	–	29	71.3	62.1
UNCLASSIFIED	4,596	66.8	48.0	162	45.9	39.5

Encoder versus raw values. Table 17 compares LagFormer ($C = 24$) with the same model on raw values alone ($C = 1$), per series and averaged over three seeds of each. The encoder does not change which multiple of the period is found: on real series the two agree with the catalogue equally often. What it changes is where the estimate lands. On the series both models place within 10% of the reference, the encoder's median error is 7% lower on synthetic series and 9% lower against the catalogue period on real ones, and it is the closer of the two on 56–57% of series. The accuracy gain on synthetic series is largest at low signal-to-noise: $+1.2$ points $[+0.8, +1.5]$ in the lowest SNR quartile against $+0.4$ $[+0.3, +0.6]$ in the highest. On real series its sign follows fold quality, $+0.8$ $[+0.1, +1.5]$ in the worst-folding quartile ($\theta > 0.46$) and -1.0 $[-1.6, -0.5]$ in the cleanest ($\theta < 0.15$).

Table 17: Encoder ($C = 24$) against raw values alone ($C = 1$), three seeds each. Median error is $|\log_{10} \hat{P} - \log_{10} P|$ over the series both models place within 10% of the reference (true period on synthetic, catalogue period on real). Ratio: encoder over $C = 1$, with a 2,000-resample paired bootstrap interval. $\Delta\mathrm{Acc}_{10}$: paired difference over all series.

		Median error (10^{-4})			Encoder	
	n	Encoder	C=1	Ratio	closer (%)	$\Delta\mathrm{Acc}_{10}$
Synthetic	50,432	7.47	8.04	0.930 [0.923, 0.937]	55.7	$+0.69$ $[+0.57, +0.81]$
Real	11,983	6.99	7.67	0.912 [0.898, 0.927]	56.7	$+0.00$ $[-0.32, +0.33]$

H Qualitative Examples

Each page shows one family: a synthetic series with known period (top three rows) and a held-out real series (bottom two rows). Both are chosen by a fixed rule (`select_qualitative.py`), not by inspection. The synthetic series is the one, among up to 160 random test series of the family on which the first stage is correct and the contraction ends within 10^{-5} of the true period, whose correlogram and comb point at P and not at $P/2$ or $2P$, and whose folded correlogram peak stands highest above its scatter. The real series has a catalogue fold that is loose but not empty ($0.30 \leq \theta \leq 0.60$). Where possible, our full estimate is twice the catalogue period and folds at least 0.05 tighter, which happens on 56 held-out series and for 6 of the 13 families. Folding at $2P$ separates the two halves of the cycle, which a fold at P superimposes, so a lower θ there is evidence for $2P$. For the other families the full estimate shares the catalogue harmonic, and at a shared harmonic the two periods fold alike (median reduction in θ of 0.000, 99th percentile 0.003, over the 13,814 of all 17,129 held-out series where they share it for seed 42, before the $\theta < 0.7$ condition of Table 2), so those pages show agreement rather than an improvement. θ uses 18 phase bins, as in Table 2.

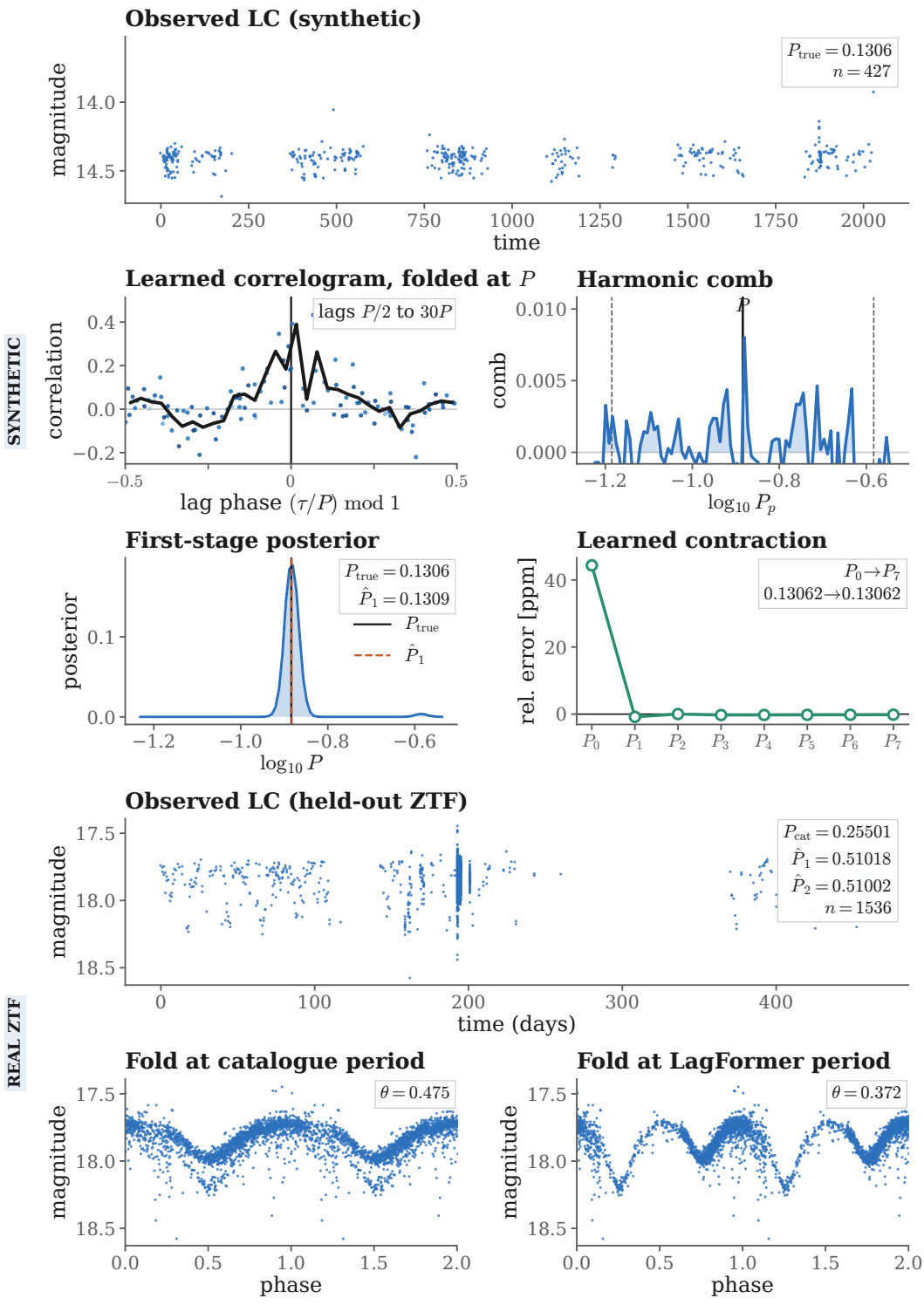

Figure 2: EW (W Ursae Majoris contact binary). *Top*, synthetic: the correlogram at every lag with pairs from $P/2$ to $30P$, folded at P so that all multiples meet at phase 0 (line: mean); the equal-weight comb; the posterior; and the contraction. *Bottom*, ZTF: our full estimate is 2.00 times the catalogue period, and θ is 0.475 at the catalogue period and 0.372 at ours.

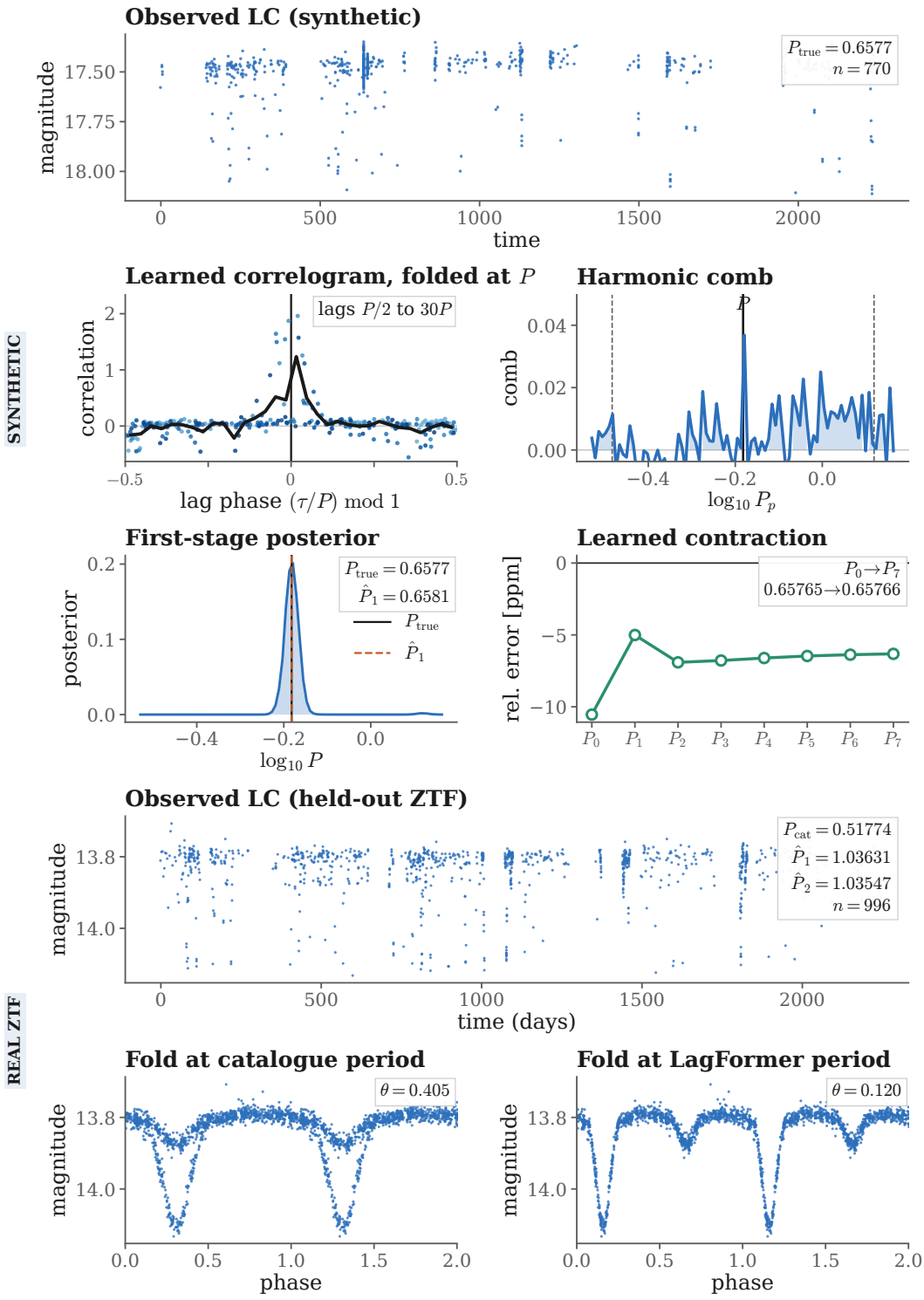

Figure 3: EA (Algol-type detached eclipsing binary). Layout as Figure 2. *Bottom*, ZTF: our full estimate is 2.00 times the catalogue period, and θ is 0.405 at the catalogue period and 0.120 at ours.

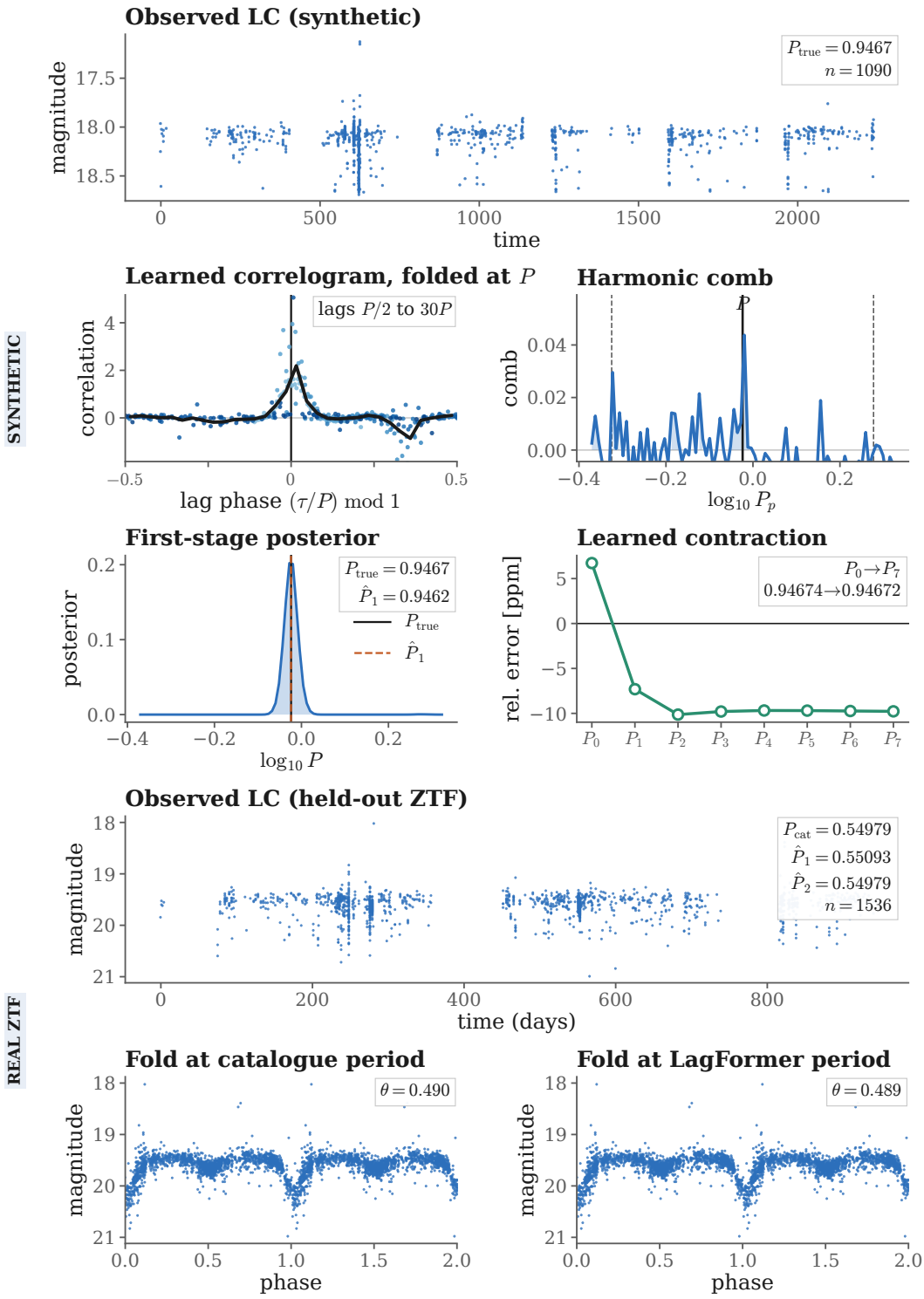

Figure 4: EB (β Lyrae-type eclipsing binary). Layout as Figure 2. *Bottom*, ZTF: our full estimate agrees with the catalogue period to 1.0 parts per million, and θ is 0.490 at the catalogue period and 0.489 at ours.

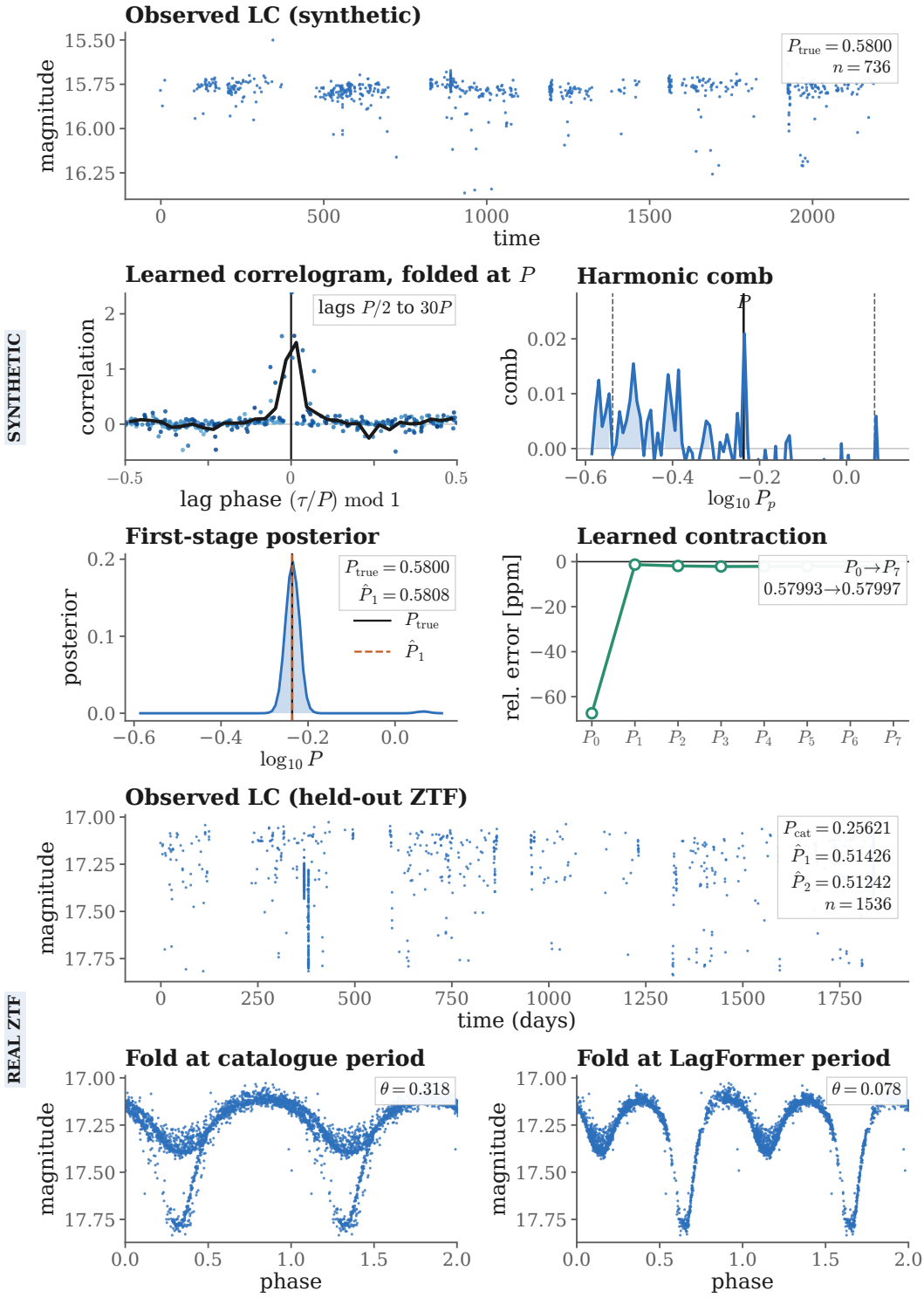

Figure 5: ECL_UNTYPED (eclipsing binary without a type). Layout as Figure 2. *Bottom*, ZTF: our full estimate is 2.00 times the catalogue period, and θ is 0.318 at the catalogue period and 0.078 at ours.

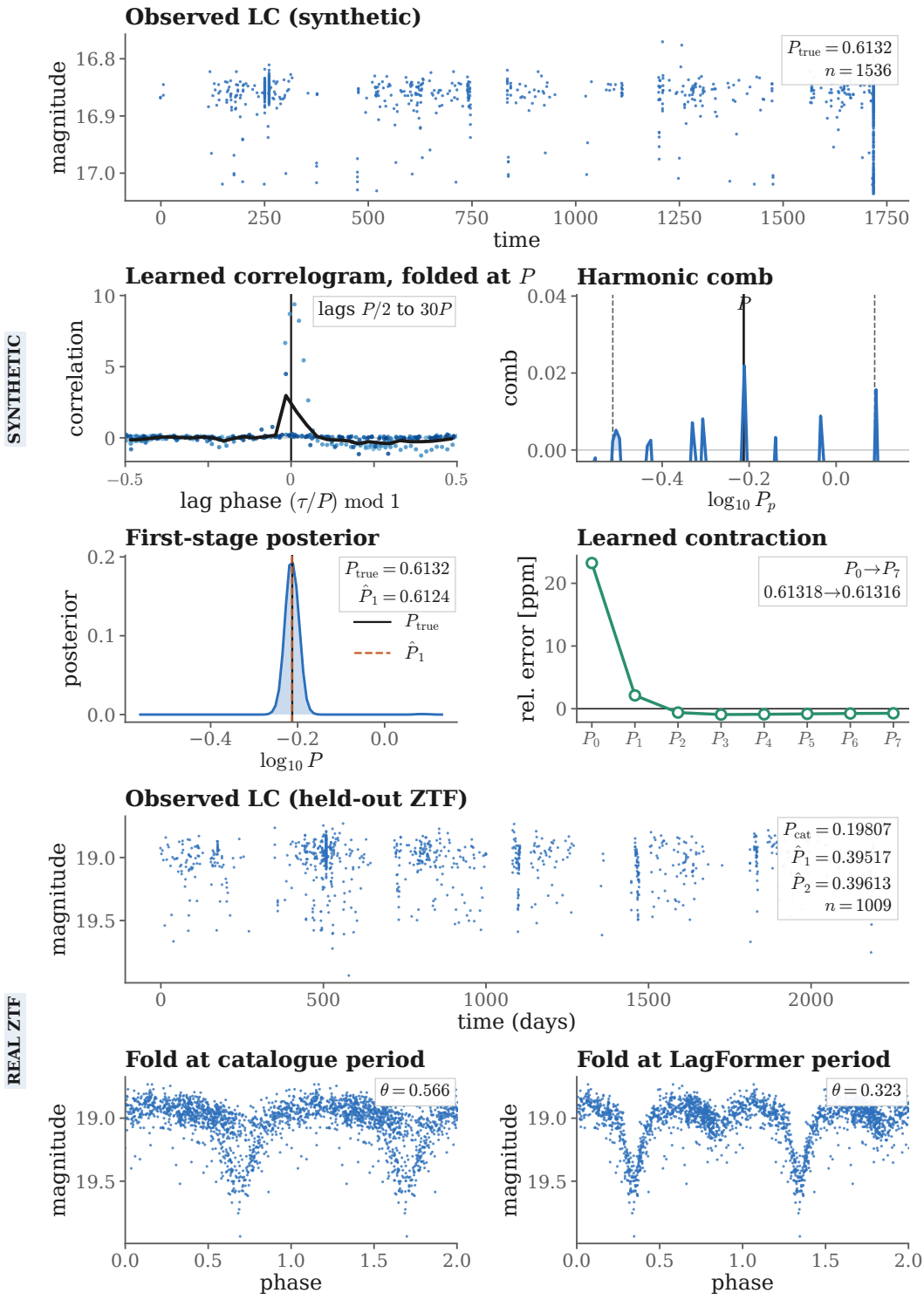

Figure 6: BIN_UNTYPED (binary without a type). Layout as Figure 2. *Bottom*, ZTF: our full estimate is 2.00 times the catalogue period, and θ is 0.566 at the catalogue period and 0.323 at ours.

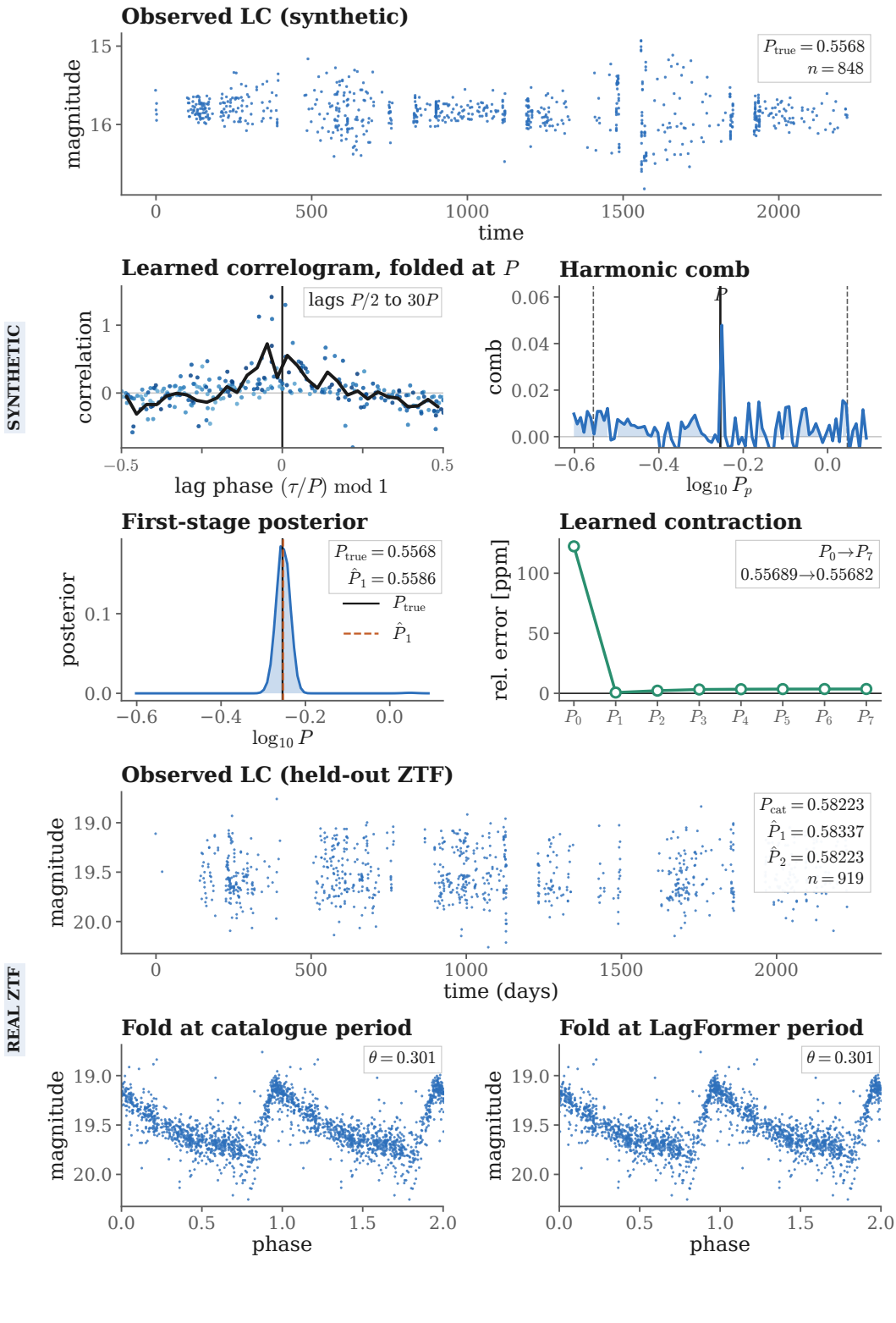

Figure 7: RRAB (RR Lyrae, fundamental mode). Layout as Figure 2. *Bottom*, ZTF: our full estimate agrees with the catalogue period to 3.1 parts per million, and θ is 0.301 at the catalogue period and 0.301 at ours.

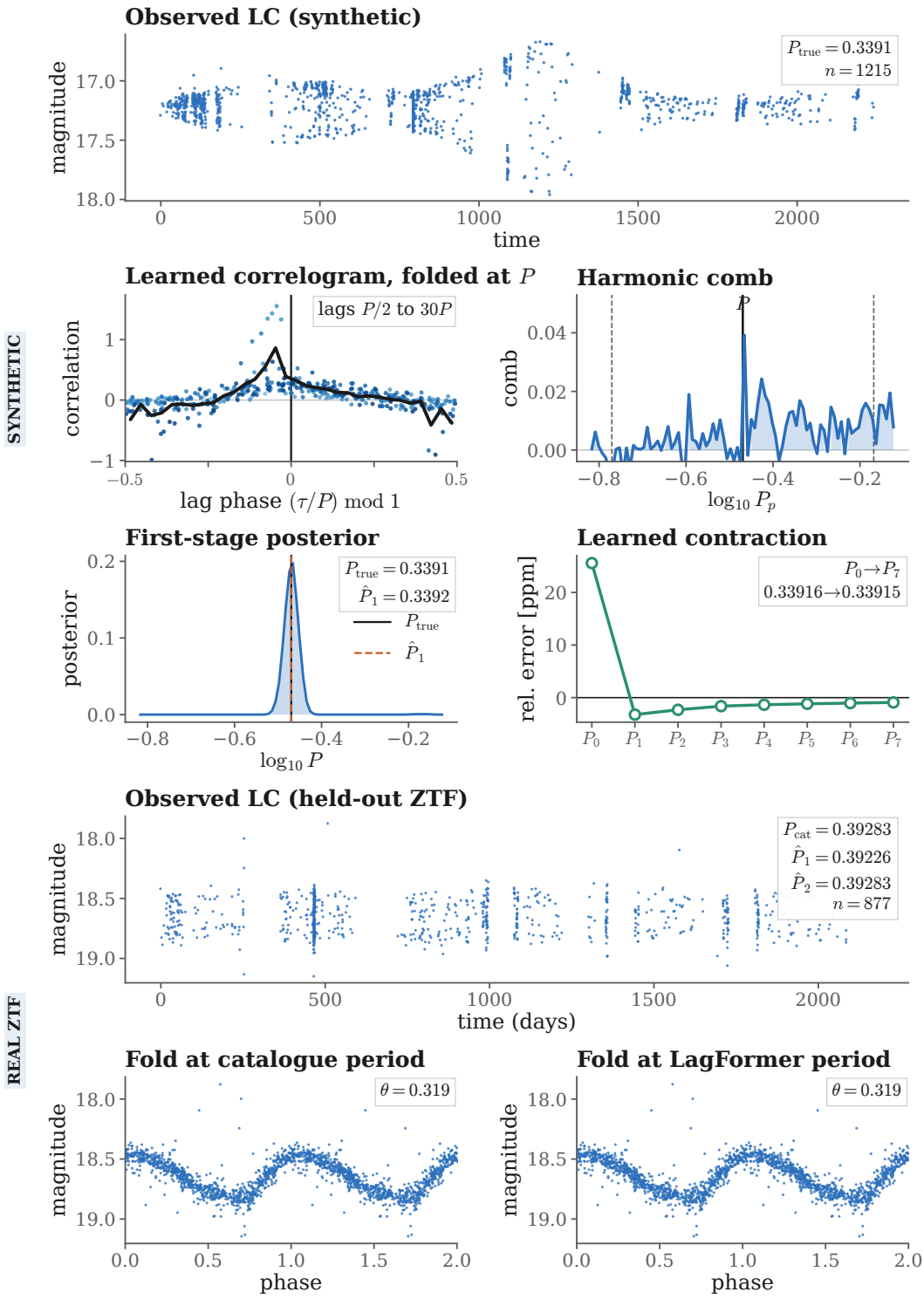

Figure 8: RRC (RR Lyrae, first overtone). Layout as Figure 2. *Bottom*, ZTF: our full estimate agrees with the catalogue period to 1.1 parts per million, and θ is 0.319 at the catalogue period and 0.319 at ours.

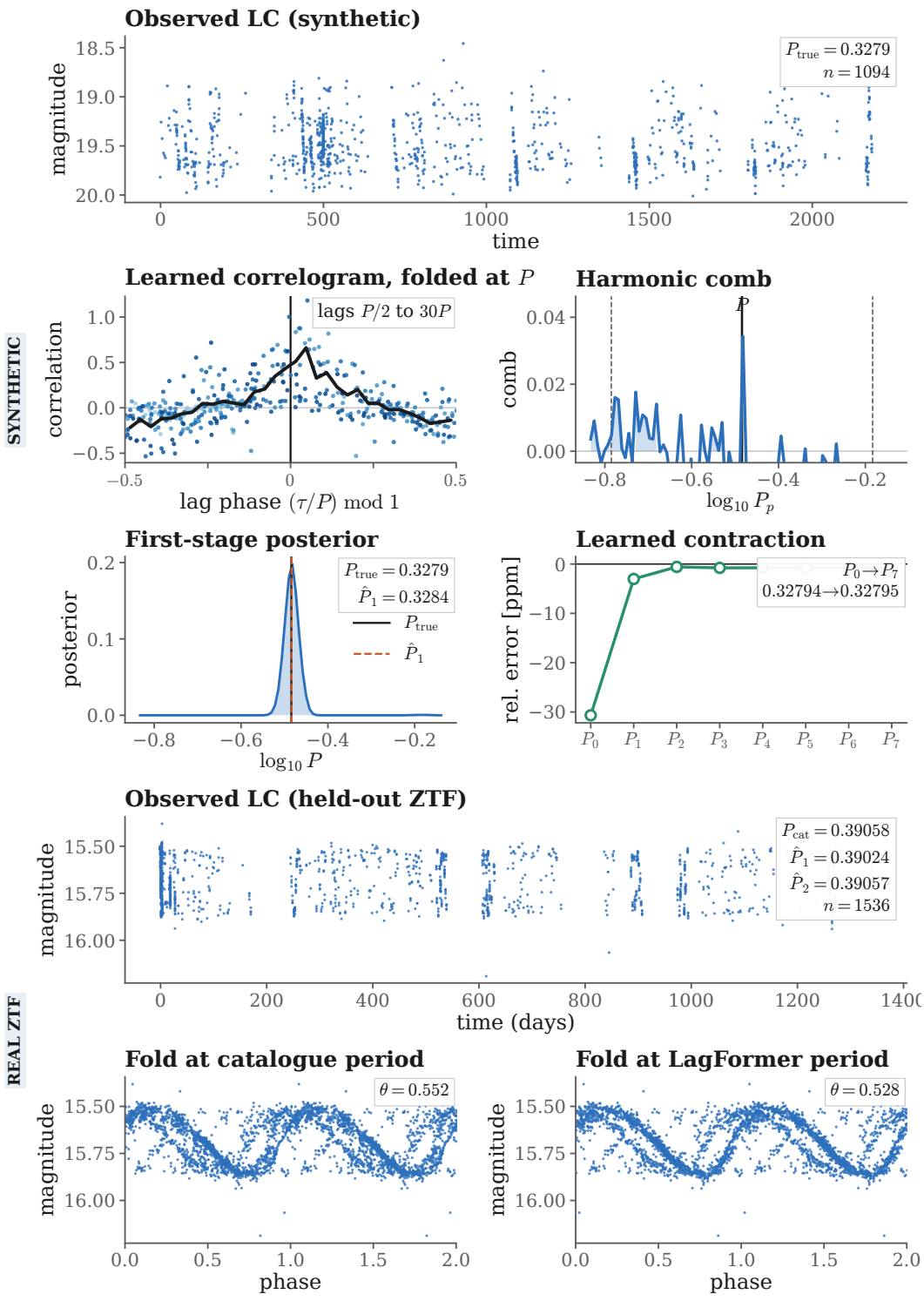

Figure 9: RRLYR_UNRES (RR Lyrae, mode not resolved). Layout as Figure 2. *Bottom*, ZTF: our full estimate agrees with the catalogue period to 28 parts per million, and θ is 0.552 at the catalogue period and 0.528 at ours.

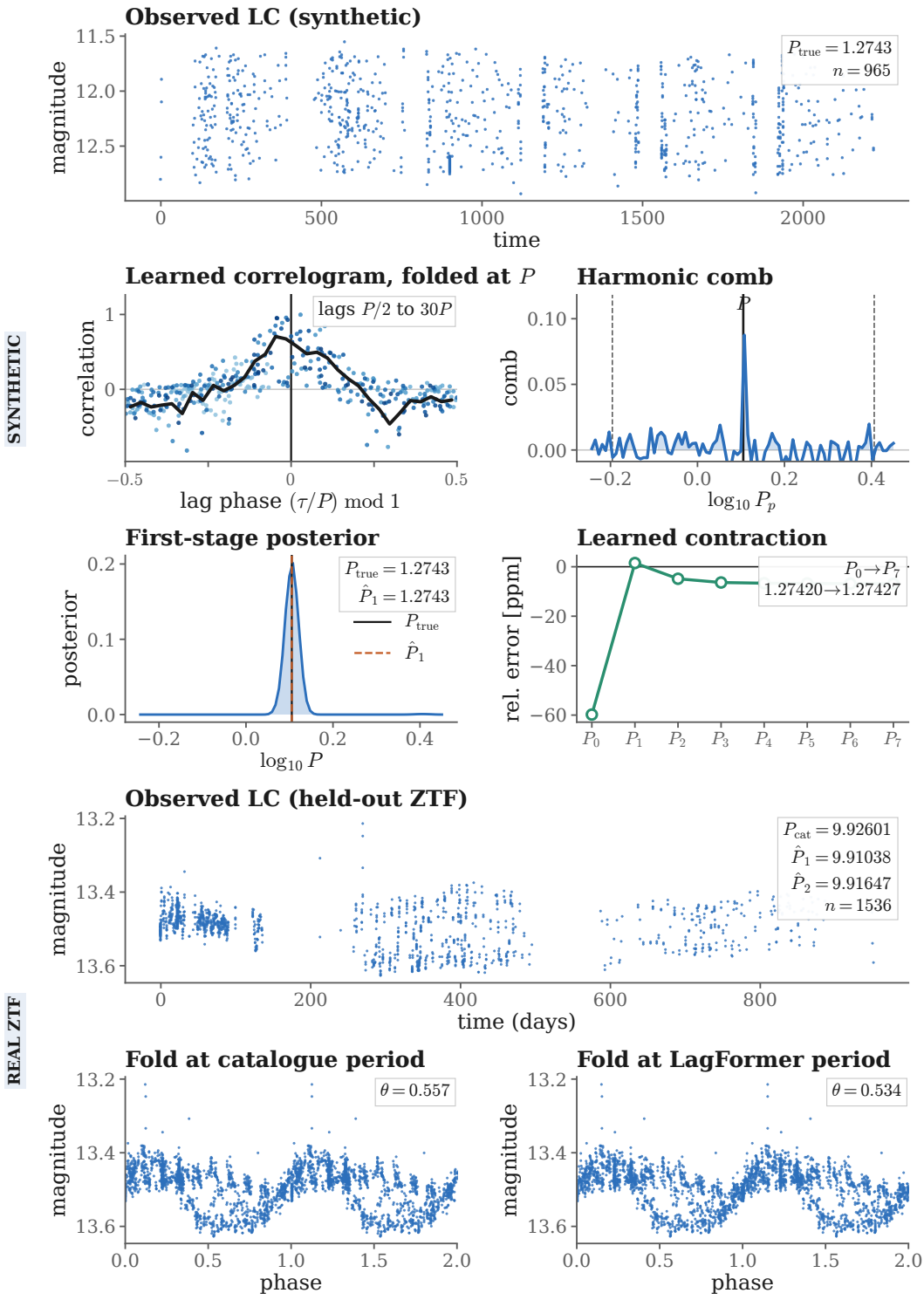

Figure 10: CEPH (Cepheid). Layout as Figure 2. *Bottom*, ZTF: our full estimate agrees with the catalogue period to 961 parts per million, and θ is 0.557 at the catalogue period and 0.534 at ours.

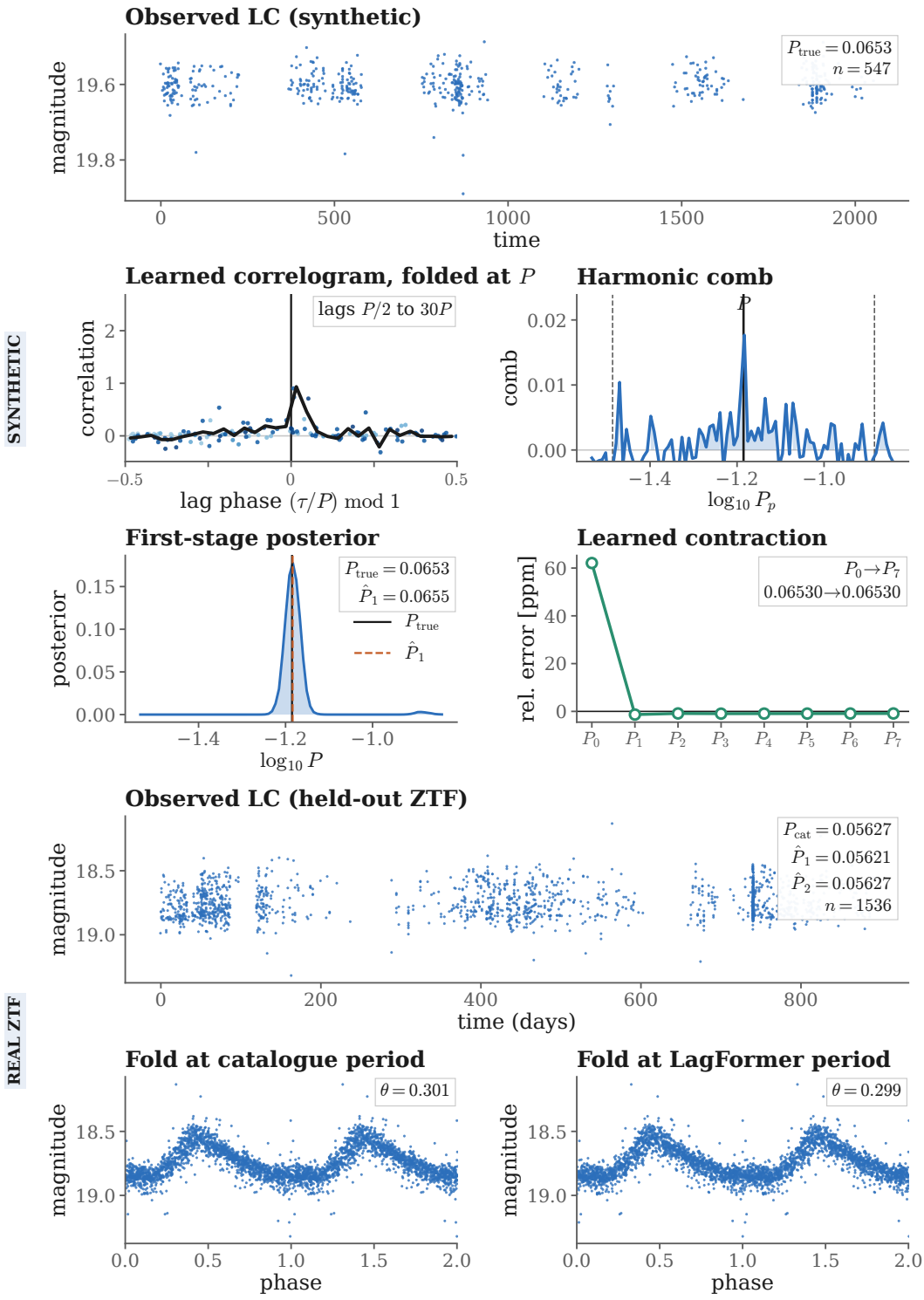

Figure 11: DSCT (δ Scuti). Layout as Figure 2. *Bottom*, ZTF: our full estimate agrees with the catalogue period to 2.1 parts per million, and θ is 0.301 at the catalogue period and 0.299 at ours.

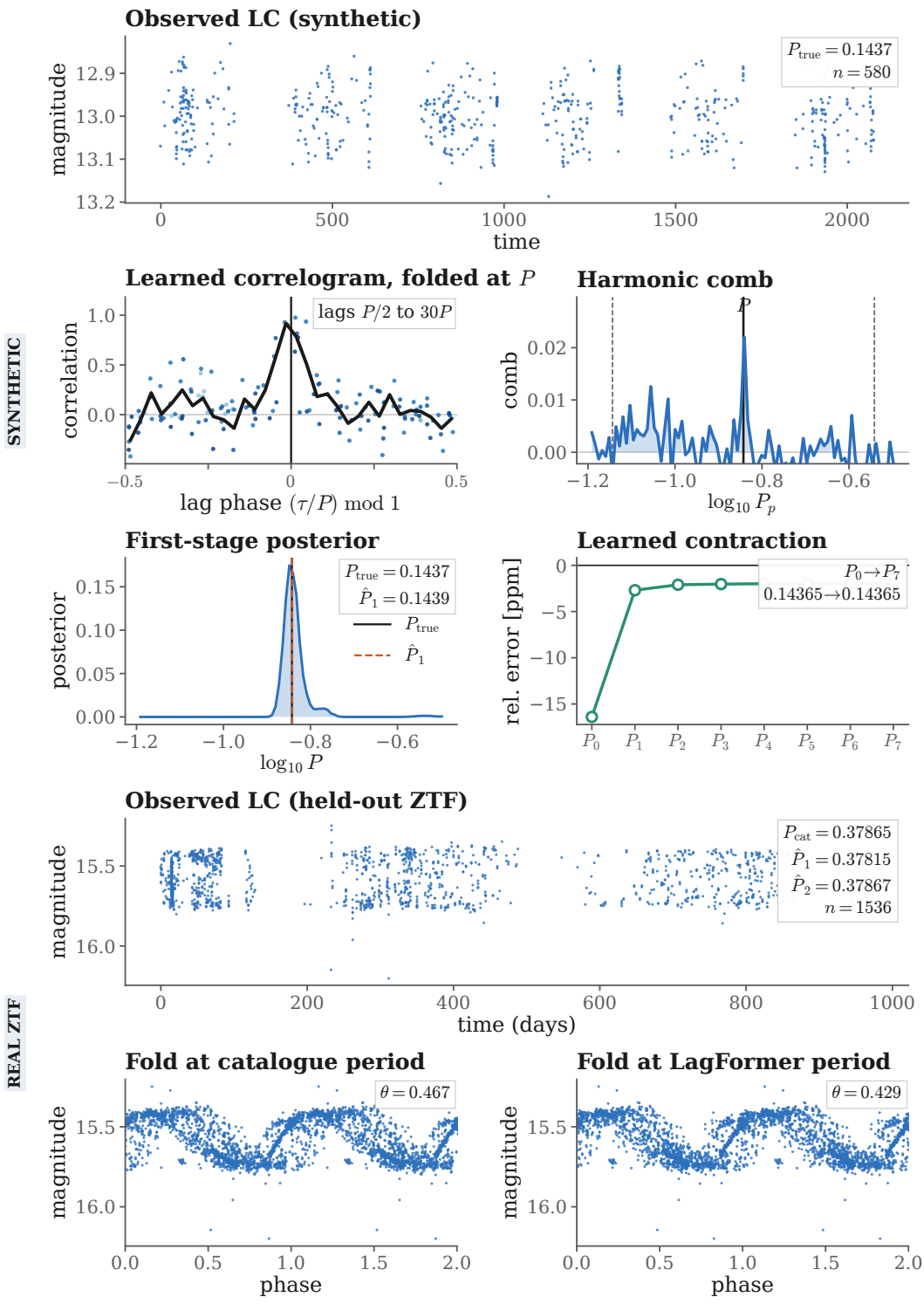

Figure 12: PULS_UNTYPED (pulsator without a type). Layout as Figure 2. *Bottom*, ZTF: our full estimate agrees with the catalogue period to 50 parts per million, and θ is 0.467 at the catalogue period and 0.429 at ours.

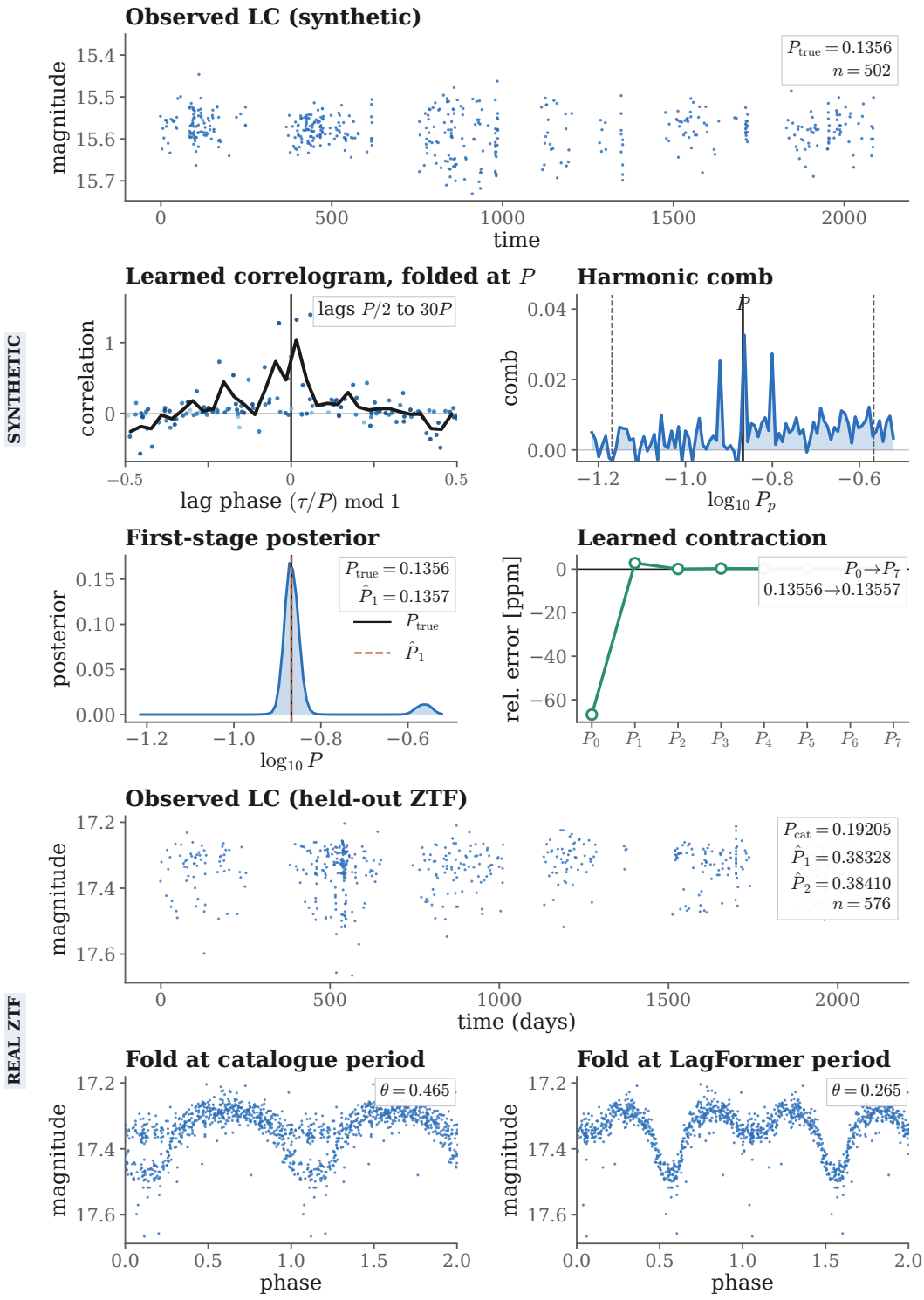

Figure 13: PNP_UNTYPED (periodic, without a type). Layout as Figure 2. *Bottom*, ZTF: our full estimate is 2.00 times the catalogue period, and θ is 0.465 at the catalogue period and 0.265 at ours.

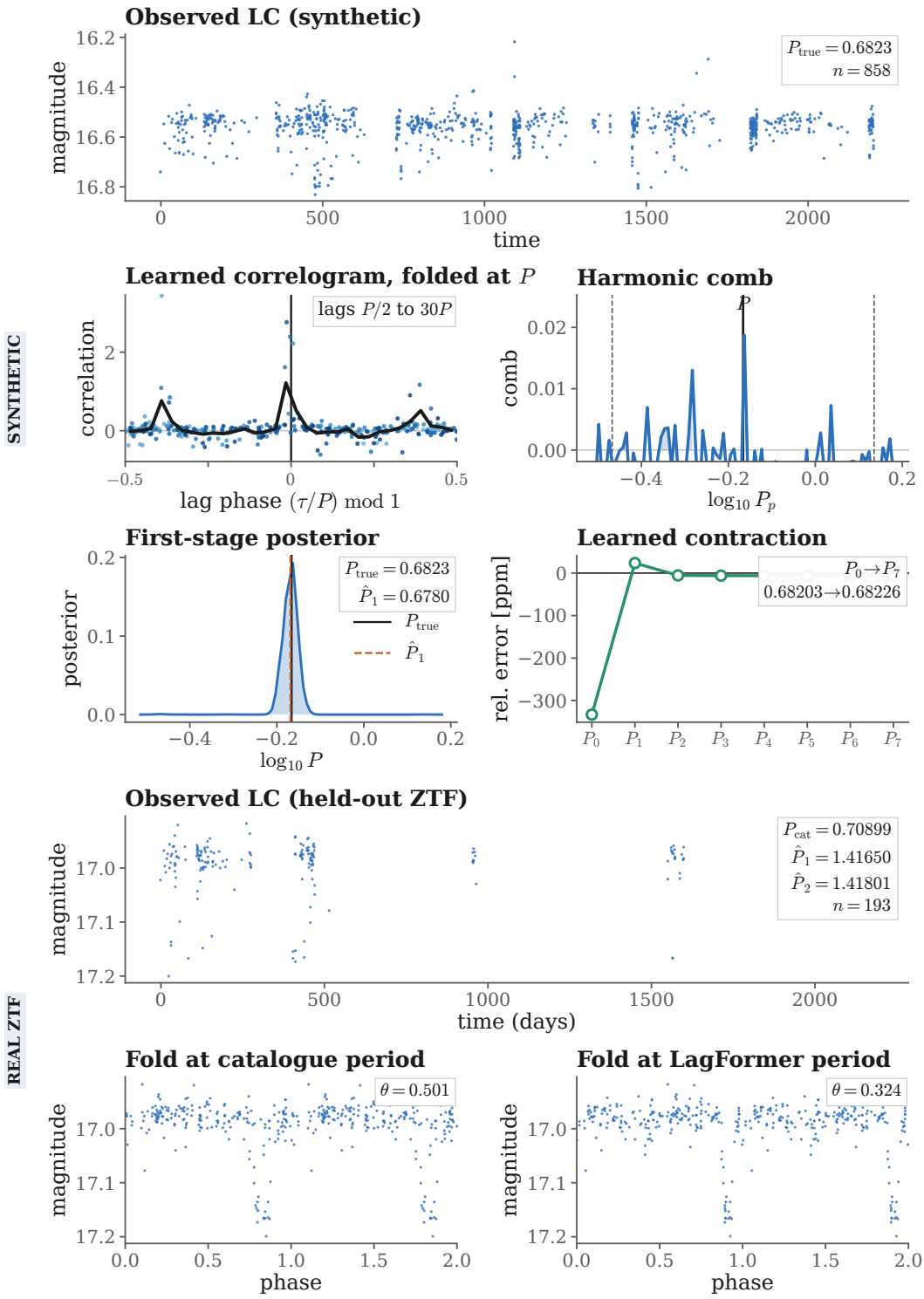

Figure 14: UNCLASSIFIED (periodic, no family in the catalogue). Layout as Figure 2. *Bottom*, ZTF: our full estimate is 2.00 times the catalogue period, and θ is 0.501 at the catalogue period and 0.324 at ours.

I SYNTHETIC CORPUS

The synthetic corpus of Section 4 is made with the generator of Anonymous (2026), a companion paper under review that builds and tests it on light curves from a different survey. This section says what the generator does, what we changed to run it on ZTF, and the settings chosen for each family.

How a series is made. Each synthetic series is built from three real pieces and a small set of settings.

1. *A shape.* The waveform of one real star, fitted as a smooth function of phase. Almost all shapes are Fourier series; for long-period stars and shapes that need more than 16 harmonics, a Gaussian process (Rasmussen & Williams, 2006) is also tried and kept if it fits better.
2. *A period and an amplitude.* The period is drawn from the family's own period distribution. The amplitude is drawn from the family's own amplitude–period relation.
3. *A donor.* The observation times and error bars of a real ZTF series. The shape is painted onto these times, and noise at the donor's own level is added.
4. *Six settings* that make the series less ideal, as real stars are: amplitude changes over time (η), slow correlated noise (ρ) with a correlation time τ, drift of the phase (δ), a second pulsation mode (β, used for DSCT only), and a spread h that lets these settings vary from series to series.

The period used to paint the series is its only label.

How the settings are chosen. The settings are not tuned by hand. For each family, the generator measures four quantities on the real stars and on synthetic series, using the same code for both: *fold quality* $\bar{\theta}$ (how cleanly the series folds at its period; lower is cleaner), *residual scatter* s (what is left after the shape is removed), *noise-to-amplitude* w, and *recovery* R (how often a standard period search finds the period or a simple multiple of it). It then searches a grid of settings and keeps, for each family, the one that (i) folds no more cleanly than the real family, with a small margin; (ii) keeps recovery at least $0.9\times$ the real value, so the period is still in the data; and (iii) is otherwise closest to the real family. The choice is then checked on a fresh draw of 2,000 series per family. In short, each synthetic family is made *no easier* than the same family in the survey.

What we changed for ZTF. The generator's code, measures, selection rule and setting grids are unchanged. The parts that read the survey were adapted, because ZTF (Bellm et al., 2019) sees a star about once a night for about 2,200 days, while the original survey sees it every two minutes for 25 days. Tables 18–22 list every fixed setting of the run, one table per stage, and mark the ones we changed. Three changes need a reason.

- *A finer period for each real star.* With the coarse step alone, the phase drifts by up to 0.1 cycle over the ZTF baseline, which blurs narrow eclipses in the fitted shapes.
- *A star joins its family only if it folds.* The same rule is applied to every real-side number the generator reads, so it is calibrated to the same stars it copies. Without it, the target spread of fold quality for ECL_UNTYPED would be 0.58 instead of 0.24.
- *Noise from same-night pairs.* The original method differences consecutive points at a regular cadence, which ZTF does not have. It is undefined for 44% of ZTF stars.

The held-out real stars of Section 4 are excluded from the shapes, the donors, and every statistic the generator reads.

Table 18: Cleaning. A point is removed only if it is far from a model of the star and nothing confirms it. Rows marked † were changed for ZTF; all other settings in Tables 18–22 are as in Anonymous (2026).

Setting	Value
Point limits†	error bar < 1.0 mag; $8 < m < 23$; at least 50 points left
Flag threshold, with / without a period	4.5 / 6.0 times the local scatter
Variance needed to trust a period	20%
Phase-model nodes; bandwidths†	256; $\{0.012, 0.025, 0.05, 0.10\}$ in phase
Bandwidth penalty; reweighting; iterations	$3/n_h$; Tukey biweight (Beaton & Tukey, 1974) at 6 MAD; 2
Phase-error scale†	0.008
Reference without a trusted period†	weighted mean of 4 neighbours on each side
What clears a flagged point†	a same-sign excursion on the same night (≤ 0.3 d)
Always removed†	bad flux or S/N; outlying error bar; level shifts

Table 19: Period, fold quality, recovery and noise. T is the baseline of a series and n_{cyc} the number of cycles it covers.

Setting	Value
Period refinement†	$\pm \min(40, n_{\mathrm{cyc}}/2)$ steps of $1/(5T)$, then $1/(50T)$
Phase bins; bin origins for $\bar{\theta}$	15; 96
Family membership†	label and $\bar{\theta} \leq 0.70$
Recovery search range; step	$0.4P$–$2.5P$; $1/(4T)$
Recovery search size (max.)†	400,000 trial frequencies
Multiples counted; tolerance; bar	$\{\frac{1}{3}, \frac{1}{2}, 1, 2, 3\}$; 10%; $0.9\times$ real
Real series per family for the bar	up to 1,200
Amplitude bins; minimum per bin	$25 \times 2^{0,\ldots,7}$; 5
Noise-floor constant; detection gate	3.0; amplitude $\geq 3\times$ the noise of a bin mean
White-noise estimate†	same-night pairs (≤ 0.3 d), at least 30 per star

Table 20: Fitted shapes.

Setting	Value
Fourier orders; ridge	$\{2, 4, 6, 8, 12, 16\}$; 10^{-4}
Cross-validation	5 contiguous time blocks
Gaussian process tried for†	long periods; shapes that need order 16
Gaussian-process cells (fit / predict); restarts	200 / 600; 2
Scatter bins	$\mathrm{clip}(\lfloor N/100 \rfloor, 40, 200)$
Fallback	40-bin median if $\chi^2_{\mathrm{red}} > 5$ or coverage < 0.8
Quality check	coverage ≥ 0.8 of 25 bins, S/N ≥ 3, $\chi^2_{\mathrm{red}} \leq 5$
Shape grid; normaliser	200 points; 99.5th minus 0.5th percentile

Table 21: Painting a series.

Setting	Value
Period distribution	density in $\log P$, Scott bandwidth (Scott, 1992) $\times 1.3$, padded 0.2 dex
Period range†	0.02–800 d
Amplitude range	family's 0.5–99.5th percentile, redrawn if outside
Shape depth gate	peak-to-peak $\geq 2\times$ the real scatter
Shape warp	clipped to ± 0.4
Donor candidates	25 nearest in white-noise level
Amplitude change	4 sinusoids, 0.5–2.5 cycles per series
Second modes (DSCT)	2–3; offsets $\mathcal{U}(0.05, 0.40)$; ratios $\mathcal{U}(0.45, 0.95)$
Glitch amplitude; rate†	$\pm\mathcal{U}(4, 8)\times$ noise; share removed by cleaning
Held-out stars†	10% of each family, excluded everywhere

Table 22: The search. Each family's six settings are chosen from these grids by the rule above; Table 23 gives the values chosen.

Setting		Values
Amplitude change	η	0, .05, .10, .15, .20, .30, .45, .60, .80, 1.0, 1.3
Slow correlated noise	ρ	0, .02, .05, .10, .15, .25, .40, .60, .85
Its correlation time	τ/T	.003, .008, .02, .05, .15
Phase drift	δ	0, .01, .03, .06, .12, .20
Second mode (DSCT only)	β	0, .35, .70, 1.0
Series-to-series spread	h	0, .4, .8, 1.2, 1.6, 2.0
Series per setting; margin†		600; 0.02 (0.05 for RRC)
Confirmation†		24 best settings, 2,000 new series each
Series per family; split†		46,154; 80/10/10 by shape

Settings for each family. Table 23 lists, for each of the 13 families, the real stars behind it and the chosen settings. Table 24 gives the check on the generated series. The *Folds* column shows how hard each family is at ZTF depth: 97% of RRAB stars fold cleanly, but only 8% of DSCT stars. Every family meets every

requirement: fold quality is 1.02–$1.21\times$ the real value (harder), residual scatter is at least $1.08\times$ the real value, noise-to-amplitude is within 8% of the real value, and recovery is at least $0.97\times$ the real value.

The settings follow the stars. Families that mix several kinds of star (PNP_UNTYPED, ECL_UNTYPED, UNCLASSIFIED) get strong amplitude change ($\eta = 0.30$–0.45). Families with one clear mechanism (RRAB, EA, EW) get little. The short-period pulsators DSCT and RRC get a short noise correlation time ($\tau/T = 0.003$, about a week). DSCT is the only family where the second-mode setting applies, and the search turned it off ($\beta = 0$).

Table 23: Real stars and chosen settings for each synthetic family. *Stars*: real stars that pass the fold rule and are not held out. *Shapes*: fitted shapes available to the family. *Folds*: share of the family's labelled stars that pass the fold rule ($\bar{\theta} \leq 0.70$). *Period*: 10th–90th percentile of the family's period distribution, in days. η: amplitude change; ρ: share of slow correlated noise; τ/T: its correlation time as a fraction of the baseline; δ: phase drift; β: second-mode strength (DSCT only); h: series-to-series spread of the settings.

Family	Folds (%)	Stars	Shapes	Period (d)	η	ρ	τ/T	δ	β	h
BIN_UNTYPED	70.0	20,442	20,872	0.154–1.29	0.05	0.00	0.150	0.01	–	1.2
CEPH	86.0	274	300	1.86–20.7	0.00	0.00	0.150	0.03	–	0.0
DSCT	7.9	1,673	1,674	0.0475–0.0923	0.30	0.05	0.003	0.00	0.00	1.2
EA	95.5	2,949	2,991	0.363–2.25	0.10	0.02	0.008	0.01	–	0.0
EB	99.9	813	815	0.446–1.22	0.05	0.02	0.050	0.01	–	0.0
ECL_UNTYPED	11.9	17,420	17,936	0.163–5.89	0.30	0.10	0.050	0.00	–	0.0
EW	75.6	70,622	70,804	0.127–0.318	0.10	0.02	0.150	0.00	–	1.6
PNP_UNTYPED	21.9	32,402	32,899	0.136–6.21	0.45	0.02	0.150	0.03	–	0.0
PULS_UNTYPED	31.1	2,462	2,521	0.0577–75.6	0.00	0.10	0.020	0.01	–	1.2
RRAB	97.2	2,151	2,158	0.475–0.657	0.10	0.02	0.150	0.00	–	0.8
RRC	99.8	1,023	1,102	0.279–0.392	0.10	0.00	0.003	0.01	–	1.2
RRLYR_UNRES	88.7	1,847	1,887	0.35–0.687	0.05	0.02	0.150	0.01	–	1.6
UNCLASSIFIED	–	1,151	1,184	0.134–16.9	0.45	0.10	0.008	0.01	–	0.0

Table 24: Check on the generated series, 2,000 per family. $\bar{\theta}$: median fold quality (lower is cleaner), real and synthetic. The other columns are synthetic over real: s residual scatter, w noise-to-amplitude. R: recovery rate. A family is accepted when the $\bar{\theta}$ and s ratios are at least 1, the w ratio is close to 1, and synthetic R is at least $0.9\times$ real R.

	$\bar{\theta}$			s	w	R	
Family	Real	Synth.	Ratio	Ratio	Ratio	Real	Synth.
BIN_UNTYPED	0.254	0.261	1.028	1.08	1.02	0.980	0.981
CEPH	0.102	0.123	1.209	1.39	1.06	0.971	0.995
DSCT	0.426	0.470	1.104	1.20	1.01	0.987	0.966
EA	0.249	0.254	1.020	1.11	1.05	0.992	0.992
EB	0.146	0.159	1.090	1.31	1.00	1.000	0.998
ECL_UNTYPED	0.360	0.377	1.048	1.22	1.00	0.953	0.977
EW	0.206	0.223	1.087	1.21	1.00	0.992	0.964
PNP_UNTYPED	0.505	0.552	1.092	1.27	1.01	0.941	0.967
PULS_UNTYPED	0.294	0.311	1.055	1.29	1.08	0.916	0.930
RRAB	0.101	0.106	1.053	1.15	1.00	0.986	0.994
RRC	0.121	0.131	1.084	1.19	1.01	0.989	0.987
RRLYR_UNRES	0.134	0.142	1.061	1.09	1.01	0.973	0.970
UNCLASSIFIED	0.574	0.625	1.089	1.14	1.05	0.890	0.926

The delivered set. With these settings we draw 600,002 series, the same number for each of the 13 families, so the family sizes in the training data tell a model nothing. They use 98,066 distinct shapes, and the split into training, validation and test is by shape (Section 4). The set is a larger draw from the same settings, shapes and donors as the checked series of Table 24. Three families are not generated. For accretors and the sinusoidal bucket, the fitted shapes do not match the stars well (peak-to-peak depth 0.81 and 0.84 of the data). The long-period family has so few cycles per series that its statistics are unstable. The unclassified family consists of unlabelled stars that pass the same fold rule and go through the same steps.