PATHFM: FAST GENERATIVE FORECASTING OF MULTIVARIATE TRAJECTORIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series foundation models are becoming increasingly popular for general-purpose forecasting. However, most leading models predict marginal quantiles, leaving the dependence structure of future trajectories unspecified. This distinction matters for trajectory-dependent quantities such as cumulative demand, peak load, and event timing, yet point and marginal scores do not reveal it. To model this dependence, we introduce PathFM, a trajectory-based forecasting model that uses flow matching to generate complete future paths. We further distill the model to one or two steps and share history attention cache across draws. Together, these enable PathFM to generate multiple paths for a single series in roughly the time needed for one query to a quantile model. The distilled sampler also makes it practical to optimize a calibration loss directly. PathFM achieves competitive marginal accuracy on the established forecasting benchmarks fev-bench and TIME. To complement these evaluations, we introduce Forecast Path Score Bench (fps-bench), which augments fev-bench tasks with energy and variogram scores and downstream metrics. PathFM achieves the highest aggregate path score on fps-bench among the evaluated models. Experiments on chaotic systems further show that our flow model sustains long-run variability, while strong baselines rapidly collapse. PathFM makes generative time series forecasting both competitive and practical, enabling diverse downstream applications.

1 INTRODUCTION

Time series foundation models (TSFMs) are a hot topic (Ansari et al., 2024; Das et al., 2023; Ansari et al., 2025; Jain & Sen, 2026; Cohen et al., 2025). The field is growing fast: Every year, new models are released that refresh the leaderboard. The models are commonly evaluated through point accuracy (e.g., MASE) and marginal probabilistic scores (e.g., CRPS, scaled quantile loss) (Aksu et al., 2024; Shchur et al., 2025). Consequently, most of the recent TSFMs are designed and trained to produce quantile levels, which are optimized for these benchmarks.

Point accuracy and marginal probabilistic scores are widely adopted for good reason: they provide an interpretable, per-step summary of forecast quality that directly supports common operational decisions such as inventory planning, resource allocation, and risk management. However, they do not capture all aspects of forecast quality. Other properties, particularly *trajectory-dependent* ones, are important but overlooked by these metrics. Among many, we illustrate two important aspects:

- **Learning distributions with multiple modes:** Quantiles are defined through the inverse of the cumulative distribution function (CDF) of the target distribution. These CDFs, however, are not always nice and smooth, for instance, when the target distribution has multiple modes or sharp peaks. With a few or a few dozens of quantiles, the resulting approximation of the CDF can be quite rough and suffer from a regression-to-the-mean issue (Yu et al., 2026). See the Lorenz oscillator example in Figure 1.
- **Learning autocorrelation:** Many quantile regression models predict marginal quantiles at individual time steps across a forecast window. These marginal outputs do not specify temporal dependence among future values within that window. For instance, many forecasting problems also require learning trajectory-dependent quantities, such as cumulative demand, peak load, and threshold-crossing times (Perez-Diaz et al., 2025; Cai et al., 2026). See the solar-output example in Figure 1. A marginal quantile loss does not identify these joint properties.

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

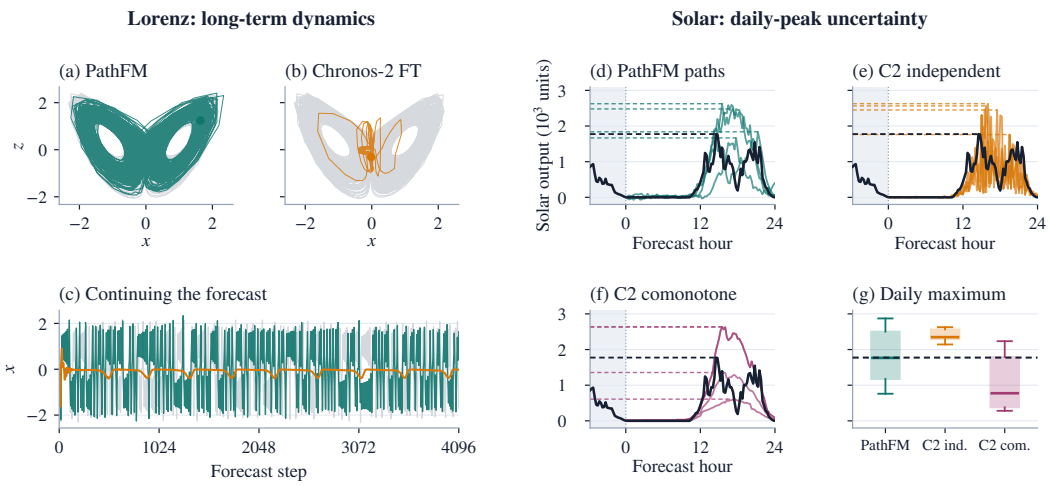

Figure 1: Two advantages of trajectory generation. Left: a sampled Lorenz rollout sustains variability and a two-lobe structure, while the fine-tuned Chronos-2 trace contracts. Both use the dynamical models in Section 6. Gray Lorenz curves show the observed continuation. Right: solar-output paths from PathFM and independent (ind.) and comonotone (com.) completions of the Chronos-2 (C2) quantiles. Horizontal dashed lines marks each solar path's daily maximum; black denotes the observed future. In (g), boxes show medians and interquartile ranges across 32 paths, whiskers show the 10–90% range, and the dashed line marks the observed maximum.

Ad hoc methods have been proposed to mitigate these weaknesses, such as completing marginal quantiles under independence or comonotone assumptions (Perez-Diaz et al., 2025), or learning copula structures on top of marginal forecasts (Drouin et al., 2022; Ashok et al., 2024). On the other hand, diffusion- or flow-matching-based generative time-series models are naturally well-suited for capturing joint trajectory distributions even without these ad hoc methods, as they can generate complete future paths while maintaining temporal and cross-variable dependencies. Unfortunately, existing generative models (Kollovieh et al., 2025; Liu et al., 2025) achieve neither high marginal accuracy nor efficient sampling, limiting their practical utility in real-world forecasting scenarios. This leads us to ask:

Can we construct a generative time-series model that achieves both high marginal accuracy and efficient sampling while also preserving trajectory dependence?

We introduce PathFM to combine joint and multi-mode trajectory modeling with accurate marginal forecasting and efficient sampling. The model learns temporal and cross-variable dependence through conditional flow matching (Lipman et al., 2022), generating complete future paths conditioned on history and known future covariates. Parallel decoding distillation (Shaul et al., 2026) and history attention caching enable faster sampling with one or two model evaluations. To narrow the marginal accuracy gap, we add a calibration loss that directly optimizes fair CRPS on generated forecasts. The distilled sampler makes this optimization practical by allowing calibration gradients to pass through its short rollout and refine the forecast distribution.

With these components, PathFM ranks *third* on the widely used fev-bench for marginal scores (Section 5). On Forecast Path Score Bench (fps-bench), which we curate to evaluate trajectory quality, PathFM ranks *first* for path scores (Section 5). In a separate evaluation on chaotic systems, our sampling approach improves point forecast accuracy upon the state-of-the-art PANDA model, captures useful temporal dependence, and preserves long-run variability more faithfully than baselines (Section 6). These results demonstrate that PathFM generates high-fidelity trajectory forecasts without sacrificing marginal accuracy, making it a strong candidate for forecasting applications.

Our contributions are as follows:

- **Fast and calibrated trajectory generation.** We introduce PathFM, a joint flow foundation model for fast, calibrated trajectory generation with accurate small-ensemble marginals (Figure 2).

Distillation enables one- or two-step generation, marginal-CRPS refinement improves empirical coverage, and stratified sampling reduces finite-ensemble marginal error.

- **A benchmark for trajectory distributions.** We introduce fps-bench, to our knowledge the first dedicated benchmark for evaluating path distributions from time series foundation models on real-world data. It reveals how strong marginal accuracy in existing models can mask errors in temporal dependence and uncertainty over trajectory-dependent quantities.
- **State-of-the-art performance.** PathFM outperforms leading foundation models on downstream probabilistic forecasting tasks and stays among the top on marginal benchmarks. It also achieves the highest forecasting accuracy across strong baselines on chaotic systems.

2 Problem Setup and Preliminaries

Forecast object and downstream query. Let $H \in \mathbb{R}^{T\times C}$ be the observed history and $Y \in \mathbb{R}^{L\times C}$ the future. The available information $\mathcal{I}$ includes H, observation masks, and known future covariates. A predictive law $P_\theta(Y \mid \mathcal{I})$ supports queries about individual coordinates or a trajectory functional $g(Y)$, such as cumulative sum or autocorrelation. A trajectory generator represents this law through $X = G_\theta(\mathcal{I}, Z)$, where each noise draw Z produces one complete future path.

Perez-Diaz et al. (2025) distinguish point, quantile, parametric-distribution, and trajectory forecast products by the decisions they support. Coordinatewise quantiles $Q_\tau(Y_{\ell,c} \mid \mathcal{I})$ and marginal distribution parameters leave temporal and cross-variable dependence unspecified. Consequently, they do not determine predictive autocorrelation or distributions of sums, maxima, and path-dependent events without an additional dependence assumption. Joint trajectory samples support direct estimation of those quantities by applying g to each generated path. Table 1 compares forecast objects, computational costs, and task compatibility; Section A discusses related work.

Table 1: Forecast objects, inference cost, and task compatibility of representative models.

Model	Forecast object	NFE	Task compatibility			
			Pretrained	Marginal	Path	Cross-variable
Quantile models						
Chronos-2 (Ansari et al., 2025)	Marginal quantiles	1	✓	✓	×	×
TimesFM-3 (Jain & Sen, 2026)	Marginal quantiles	1	✓	✓	×	×
Toto 2.0 (Khwaja et al., 2026)	Marginal quantiles	1	✓	✓	×	×
Flow models						
TSFlow (Kollovieh et al., 2025)	Joint paths	32	×	✓	✓	✓[†]
Sundial (Liu et al., 2025)	Univariate paths	50	✓	✓	✓[‡]	×
PathFM (Ours)	Joint paths	1–2	✓	✓	✓	✓

Path and *Cross-variable* refer to native joint forecast laws. NFE counts evaluations per output block (per sampled path when applicable). [†]TSFlow's multivariate extension adds attention across features. [‡]Temporal coupling here refers to the block generated by the conditional flow.

Existing benchmarks and their scope. GIFT-Eval reports point-forecast metrics and approximates univariate CRPS with weighted quantile loss (Aksu et al., 2024); fev-bench uses scaled quantile loss across heterogeneous forecasting tasks (Shchur et al., 2025). TIME evaluates zero-shot forecasting on freshly curated datasets, reporting MASE and quantile-based CRPS (Qiao et al., 2026). These summaries assess central predictions and marginal uncertainty, including on multivariate tasks, but cannot distinguish joint laws with identical coordinatewise marginals. We introduce fps-bench to address this gap by evaluating temporal dependence and trajectory-dependent quantities (Section 4).

Section C.1 analyzes finite-ensemble approximation, and Sections D.1 and F define the evaluation metrics for general-corpus and chaotic-system prediction, respectively.

3 Fast and Calibrated Trajectory Forecasting

PathFM learns a flow over complete future trajectories, then distills it for fast sampling and refines its marginal scores. Figure 2 summarizes the training and sampling stages.

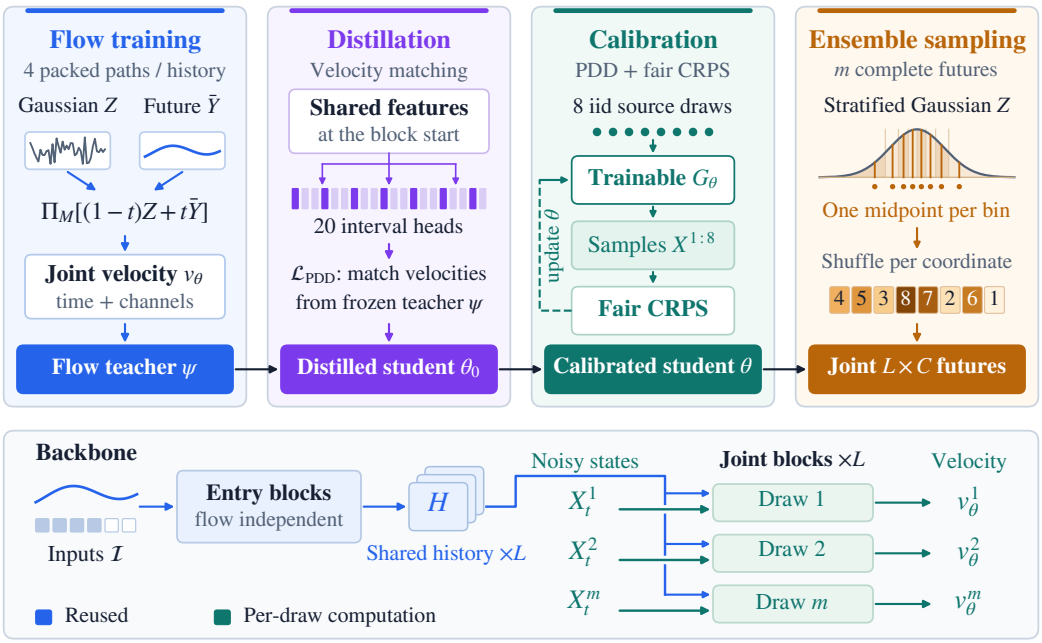

Figure 2: Training, post-training, and sampling with PathFM. Flow matching learns a joint velocity field; distillation matches frozen-teacher velocities at student-proposed states. Calibration maps eight iid draws through one fused step of trainable G_θ to samples scored by fair CRPS, together with PDD supervision. Sampling uses stratified initial noise to improve coverage of the noise distribution with a small ensemble. Below: Flow enters after the flow-independent prefix, and history attention values are computed once and reused by separate packed future states.

3.1 Pretraining by Flow Matching

Conditional flow matching (Lipman et al., 2022) learns a velocity field that transports source noise toward the conditional law $P_\theta(Y \mid \mathcal{I})$ over complete $L \times C$ futures. Here $\mathcal{I}$ contains the observed history, masks, and known future covariates. We transform targets to $\bar{Y}$ using context-derived location and scale followed by the inverse hyperbolic sine (Ansari et al., 2025). We draw Z independently of the future, with the same shape as $\bar{Y}$ and independent coordinates $Z_{\ell c} \sim \mathcal{N}(0, \sigma^2)$. The mask M selects valid unknown coordinates, Π_M restores known values and zeros invalid coordinates, and $\|\cdot\|_M^2$ averages squared error over transported coordinates.

At flow time $t \in [0, 1]$, we train v_θ on the straight path from noise to data:

$$X_t = \Pi_M\big((1-t)Z + t\bar{Y}\big), \quad \mathcal{L}_{\mathrm{FM}} = \mathbb{E}\left[\big\|v_\theta(X_t, t; \mathcal{I}) - (\bar{Y} - Z)\big\|_M^2\right]. \tag{1}$$

The expectation is over training pairs $(\mathcal{I}, \bar{Y})$, source noise, and flow time. Training uses one scalar flow time t per row, shared across all channels and forecast steps (Section B.1).

Inference integrates the ordinary differential equation, yielding $\widehat{X}_1$, one complete trajectory:

$$\widehat{X}_0 = \Pi_M(Z), \qquad \frac{\mathrm{d}\widehat{X}_t}{\mathrm{d}t} = M \odot v_\theta(\widehat{X}_t, t; \mathcal{I}), \quad t \in [0, 1]. \tag{2}$$

3.2 Efficient Backbone for Joint Generation

The velocity field uses a transformer based on T5 (Raffel et al., 2020), with separate time and channel axes. Its inputs are patches of 16 values with masks and time indices, a register token, and future patches populated by available covariates. Each block combines temporal self-attention, attention across channels at aligned patches, and a pointwise feed-forward network. We condition flow time via adaptive RMS normalization (Chen et al., 2024) and apply QK normalization and weight decay.

For reuse across paths, *late entry* separates conditioning from flow-dependent computation. The first e blocks use unrestricted attention over history, the register, and future placeholders, without source noise or flow time. These representations incorporate the requested horizon and known future covariates. Restricting history attention before this stage would exclude that future information. Future positions then receive a gated difference between flow-state and placeholder tokens. In subsequent blocks, *history-only self-attention* keeps history independent of the sampled path: history tokens attend only to history, while future tokens attend to both history and future. Flow-time modulation acts only on register and future tokens. Section B.1 specifies the configuration.

Packing and caching. Packed execution stores history once, with a separate register/future block per draw and masks preventing communication between paths. Training averages four independent source/time draws per history, giving $4\times$ flow-matching supervision at $1.5\times$ the cost. This sharing across paths is distinct from reusing history states across successive flow evaluations. At inference, the first e blocks run once per history and horizon; their outputs are reused across paths and flow evaluations. The flow model can reuse deeper history keys and values across evaluations. Section 5.3 compares accuracy and inference cost.

3.3 Post-training: distillation with calibration

Sampling a vanilla flow requires tens of sequential velocity evaluations, each depending on the state produced by the preceding step. Distillation learns a shorter rollout, i.e., one or two evaluations. This makes both ensemble generation and differentiation through generated forecasts practical.

Distillation. Parallel decoding distillation (PDD) (Shaul et al., 2026) groups consecutive flow-time intervals into blocks, predicting all interval velocities in a block from one backbone evaluation. We initialize the student from a flow teacher that remains frozen throughout post-training. We divide $[0, 1]$ into $J = 20$ intervals with knots $\tau_j = j/J$ and widths $\Delta_j = \tau_{j+1} - \tau_j$. Each interval has an affine output head $h_{\theta,j} = W_j\phi_\theta + b_j$. All heads share features ϕ_θ computed at the block's starting state and time, including nonlinear outputs and residual inputs.

For a block of b intervals starting at s, the mean velocity advances X to X^+:

$$\bar{h}_{\theta,s,b} = \sum_{j=s}^{s+b-1} \frac{\Delta_j}{\tau_{s+b} - \tau_s} h_{\theta,j}, \quad X^+ = \Pi_M\big(X + (\tau_{s+b} - \tau_s)\bar{h}_{\theta,s,b}(X)\big). \tag{3}$$

Because the heads are affine in shared features, their interval-weighted parameter sums give a single readout per block. One block of 20 intervals requires one evaluation; two blocks of 10 requires two.

For a selected interval k, summing preceding predicted displacements gives $\widetilde{X}_k$, where the teacher supplies head k's target:

$$\widetilde{X}_k = \Pi_M\left(X_{\tau_s} + \sum_{j=s}^{k-1} \Delta_j \,\mathrm{sg}(h_{\theta,j})\right), \quad \mathcal{L}_{\mathrm{PDD}} = \mathbb{E}\left[\left\| h_{\theta,k} - \mathrm{sg}(v_\psi(\widetilde{X}_k, \tau_k; \mathcal{I}))\right\|_M^2\right]. \tag{4}$$

Here ψ denotes the frozen teacher and sg stops gradients. Gradients update the student through the selected head, with the proposed state and teacher target detached. Targets are Euler velocities at student-proposed states, supplied without a complete teacher rollout. Four packed draws each supervise one interval, with the flow-training coordinate normalization (Section B.1).

Calibration. We refine the student's marginals by backpropagating the continuous ranked probability score (CRPS) through one fused sampling step. Using few paths keeps this step affordable, but empirical CRPS has a bias that can favor underdispersion (Equation (13)). Fair CRPS (Ferro, 2014) is unbiased for the marginal population score under iid sampling and finite first moments. Let $G_\theta(Z; \mathcal{I})$ denote the normalized endpoint of a fused block spanning $[0, 1]$. For m iid source draws $Z^j \sim \mathcal{N}(0, \sigma^2 I)$, set $X^j = G_\theta(Z^j; \mathcal{I})$. At coordinate (ℓ, c), the fair estimator is

$$\widehat{C}_{\ell c} = \frac{1}{m}\sum_{j=1}^{m}\left|X^j_{\ell c} - \bar{Y}_{\ell c}\right| - \frac{1}{2m(m-1)}\sum_{\substack{i,j=1 \\ i\neq j}}^{m}\left|X^i_{\ell c} - X^j_{\ell c}\right|. \tag{5}$$

Retaining PDD provides teacher supervision while the generated endpoints are optimized against observations. With $\widehat{C}_M$ the mean of $\widehat{C}_{\ell c}$ over transported coordinates, the objective becomes $\mathcal{L}_{\text{cal}} = \mathcal{L}_{\text{PDD}} + \mathbb{E}[\widehat{C}_M]$. We evaluate trajectory quality using fps-bench (Section 4) and report direct calibration diagnostics in Section B.5.

3.4 Inference: ensemble sampling

At inference, PathFM generates m complete future trajectories, from which we estimate quantiles and trajectory-dependent quantities. We stratify the initial Gaussian noise to improve coverage at a fixed ensemble size, then propagate each draw through the distilled sampler. Calibration and fair-score evaluation use iid draws; Section C.1.2 gives the construction and implementation details.

4 Evaluating Trajectory Forecasts

Trajectory forecasts should capture temporal dependence and accurately quantify uncertainty. Marginal accuracy alone cannot establish this. We introduce Forecast Path Score Bench (fps-bench) to evaluate these properties alongside marginal scores. The benchmark combines established scoring rules into a common protocol for constructing and comparing finite trajectory ensembles.

Six complementary path scores. Different downstream queries depend on different aspects of a path, motivating the complementary diagnostics in Table 9. Following the aggregation-and-transformation framework of Pic et al. (2025), we score both complete paths and transformations that expose these aspects. Energy score assesses the distribution of level paths, while differenced energy score emphasizes local changes by scoring the initial level and subsequent increments. Variogram score compares pairwise temporal contrasts to probe how values vary together across time (Scheuerer & Hamill, 2015). We then apply scalar CRPS to the horizon total, maximum, and first-passage time to assess accumulated uncertainty, extremes, and event timing. Specifically, first-passage time records the first crossing of a history-derived threshold. fps-bench evaluates 94 fev-bench tasks (Shchur et al., 2025), excluding those with NaN values that prevent computing some path scores.

Quantile-model completions. Applying these scores requires complete paths, which marginal quantile forecasts do not specify. We evaluate all models with 32 paths, retaining native samples when available. To include quantile models, we construct paths from their marginal quantiles under two explicit dependence assumptions. Sorted quantiles define a piecewise-linear inverse CDF with constant extension beyond the outer quantiles. An independent completion draws a separate quantile rank at each forecast time and channel, whereas a comonotone completion shares one rank across an item's times and channels. Both laws preserve the same underlying interpolated marginals, allowing us to examine how the dependence assumption affects trajectory scores. These two choices serve as baselines, as different dependence assumptions may yield different scores for the same marginal forecasts. Section D.1 specifies the score definitions, estimators, and aggregation.

5 Forecasting accuracy and efficiency

We evaluate forecasts through their marginal accuracy, the path quality scores, and the cost of generating an ensemble. The experiments establish where trajectory forecasts help, then examine how distillation and score refinement make those forecasts practical. Section 6 complements these experiments with a study of chaotic systems, with a particular focus on long-run dynamics.

5.1 Experimental setup

We evaluate marginal accuracy on fev-bench (Shchur et al., 2025) and TIME (Qiao et al., 2026). Baselines include TimesFM-3 (Jain & Sen, 2026), Chronos-2 (Ansari et al., 2025), Toto-1.0/2.0 (Cohen et al., 2025; Khwaja et al., 2026), and TiRex/TiRex-2 (Auer et al., 2025; Podest et al., 2026). Quantile models use explicit independent or comonotone completions, whereas Toto-1.0 and Sundial (Liu et al., 2025) retain their native sampled paths. GIFT-Eval results appear in Section B.6.1, and the full component comparisons in Section E.1.

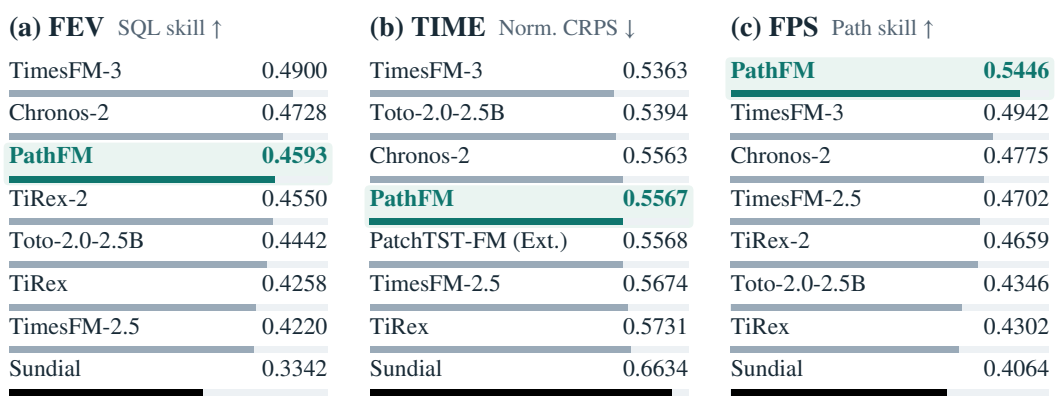

Figure 3: Marginal accuracy and trajectory quality. Native samplers PathFM and Sundial are teal and black. FPS selects each quantile model's better independent or comonotone completion.

5.2 Results

PathFM achieves the highest FPS skill among the compared models while remaining competitive on marginal accuracy (Figure 3). It ranks third on fev-bench and fourth on TIME, close to Chronos-2 on the latter. On fps-bench, it outperforms TimesFM-3 and Chronos-2 even when each quantile baseline uses its better independent or comonotone completion. It also improves on Sundial's native sampled forecasts across all three benchmarks. TimesFM-3 leads both marginal benchmarks, but its lower FPS ranking illustrates why marginal accuracy alone is insufficient to assess forecast paths.

Path-dependent scores. Figure 11 compares the six FPS components for PathFM and the three strongest external baselines. The clearest gains over TimesFM-3 (the strongest baseline) occur in scores for temporal changes, horizon totals, and maxima. These gains highlight the value of complete paths for downstream tasks that depend on temporal structure beyond marginal forecasts.

Ablation studies. We study the cumulative effects of architecture and learning choices in Section B.4. The teacher–student evaluation-budget comparison (Section B.3) and inference speedup breakdown (Figure 10) examine how distillation and shared history computation reduce sampling cost. Section B.6 provides additional results on GIFT-Eval. We also evaluate cross-variable dependence (Section E.2) and test fitted temporal copulas for quantile baselines (Section E.3).

5.3 Accuracy, model size, and inference cost

PathFM combines the highest FPS skill among the compared models with a compact 126M-parameter network (Figure 4). With one evaluation per output block, it generates 32 paths in 51.6 ms on an NVIDIA L4, placing it on the observed accuracy–latency frontier for path forecasts. We compare the number of function evaluations in Section B.3 and sampling controls in Section C.2.

Ensemble size and distillation. Figure 5 examines two ways to reduce the sampling budget in separate controlled studies. On all 100 fev-bench tasks, stratification improves marginal accuracy most at small ensemble sizes: 32-path stratified sampling achieves almost the same FEV skill as 256 iid paths. As for distillation, the student can replicate the teacher's 10-step performance with a single evaluation and remain consistent with higher NFEs. This shows the power of distillation for time series flow matching models, and the effectiveness of the parallel decoding distillation.

Scaling to 300M parameters. We also investigate the scalability of PathFM by training a 300M-parameter variant. The larger model improves both marginal accuracy and trajectory quality (Section B.6). These gains come at roughly twice the inference latency for generating 32 paths on an NVIDIA L4 (Figure 9). This shows the potential for larger flow models. All PathFM results reported elsewhere in the main text use the 126M model.

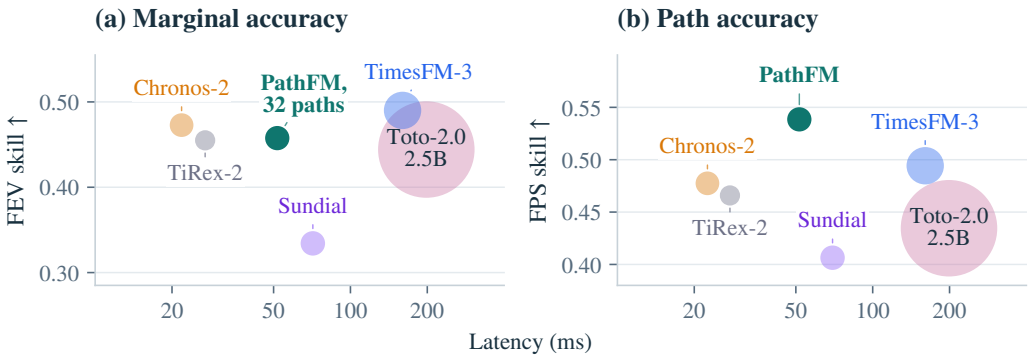

Figure 4: Accuracy, latency, and model size across six forecasting models. Area denotes parameter count. PathFM uses 32 paths and one evaluation per block. Latency is the median time to return nine quantiles (a) or 32 paths (b) per query on an NVIDIA L4, with history and forecast lengths fixed across models. Quantile models use the better completion from Section 4 for each path task.

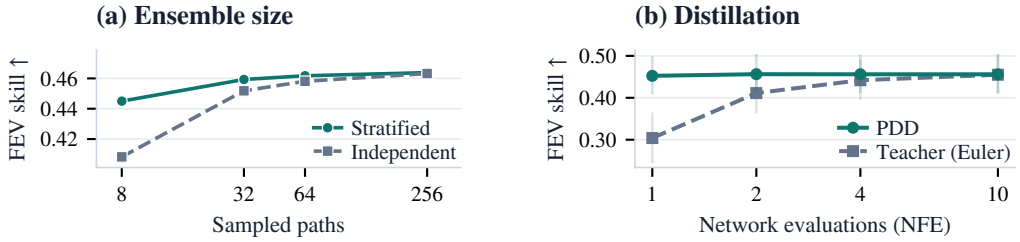

Figure 5: Ensemble size and distillation on fev-bench tasks. (a) Calibrated 126M model with different sampling strategies at NFE=1 (Section C.1.2). (b) Its matched teacher and PDD student, using 32 stratified paths. Thin bars in (b) show pointwise 95% task-bootstrap intervals (Section B.3).

6 Forecasting Chaotic Dynamics

Chaotic systems pose a sharp test of forecasting: can a model preserve statistical structure after losing alignment with a particular future? We compare fine-tuned Chronos-2, PANDA (Lai et al., 2026), and PathFM on point forecasting and long continuations of 135 DYSTS systems (Gilpin, 2021). We evaluate PathFM with ten-step Euler sampling per native 256-step block. Section F gives its training recipe, and evaluation details.

6.1 Forecast Accuracy and Dynamical Fidelity

Using sMAPE as in PANDA (Lai et al., 2026), we evaluate forecasts at horizons 64, 128, 256, and 512. Four complementary diagnostics assess continued paths: sliced Wasserstein-1 distance to an empirical approximation of the invariant distribution, correlation-dimension error, squared Hellinger distance between normalized power spectral densities (PSD), and segment analog score. Together, they assess state occupancy, geometric dimension, frequency content, and similarity of short multivariate segments to reference dynamics. Held-out truth provides a finite-trajectory reference.

It achieves substantially better dynamical fidelity over 4,096-step rollouts than all evaluated baselines (Table 2). Relative to state-of-the-art PANDA, it reduces errors across all four diagnostics by 56–74%. These gains accompany the lowest pointwise error at every reported horizon and persist when PathFM is scaled down to approximately PANDA's parameter count (Section F.2).

Chronos-1 (Ansari et al., 2024), an autoregressive forecasting model, has been shown to mimic local chaotic dynamics (Yu et al., 2026). Chronos-1 achieves lower sliced W_1 and PSD H^2 than the quantile model Chronos-2, but higher pointwise error, correlation-dimension error, and segment analog score. These results highlight the distinct strengths and weaknesses of traditional autoregressive and quantile

Table 2: Point accuracy and dynamical fidelity (lower is better; best in bold). sMAPE is the median of 480 source-group mean errors; dynamics report medians over 135 systems. PathFM uses ten steps.

	sMAPE				Dynamics			
Model	$H = 64$	$H = 128$	$H = 256$	$H = 512$	W_1	PSD H^2	CD err.	Seg. analog
Chronos-1 FT	22.94	31.42	40.95	50.44	0.279	0.130	0.666	13.72
Chronos-2 FT	17.41	24.52	31.86	39.66	0.682	0.261	0.441	10.95
PANDA	14.96	20.62	29.57	40.19	0.678	0.181	0.437	10.03
PathFM (Ours)	**12.59**	**18.08**	**25.37**	**33.39**	**0.202**	**0.047**	**0.166**	**4.42**
Held-out truth	—	—	—	—	0.025	0.006	0.036	1.00

forecasting models. PathFM improves across every reported metric, combining stronger point accuracy with greater dynamical fidelity.

Distillation does not fully retain the ten-step sampler's performance on chaotic systems. We hypothesize that sensitivity to perturbations amplifies small distillation errors during autoregressive forecasts. With one or two distilled evaluations, the margin over baselines narrows, but PathFM retains its lead. Section F.2 examines this trade-off across model sizes and sampling budgets.

6.2 What sampling preserves

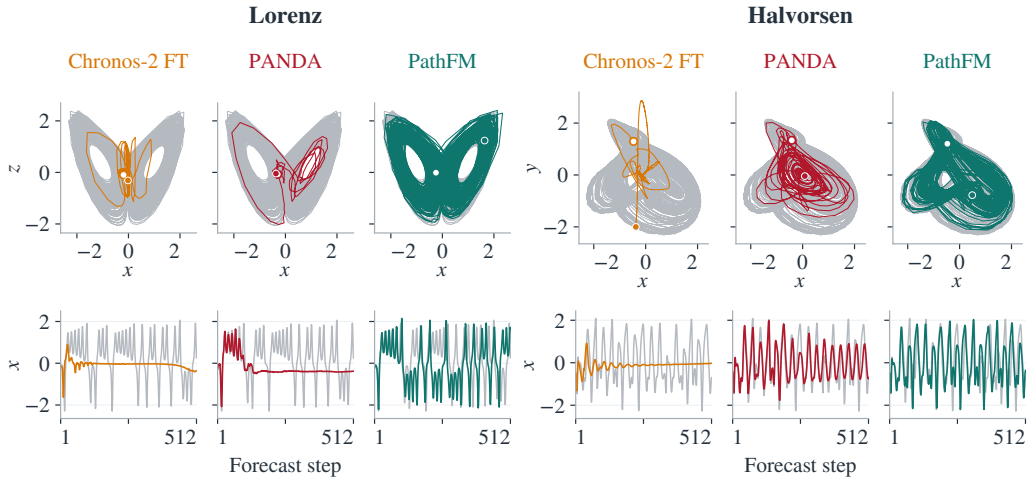

Figure 6: Lorenz and Halvorsen continuations; PathFM uses one sampled path. Each column shows a full 4,096-step portrait above its first 512 x-coordinate steps. Gray curves show the observed continuation; open and filled circles mark the first and last forecast states.

Figure 6 pairs long continuations with their early time traces for Lorenz and Halvorsen systems. The sampled Lorenz path retains a two-lobe portrait and sustained x variability, whereas the plotted baselines collapse within 256 steps. For Halvorsen, PathFM's sample retains the broad three-loop form, while PANDA follows a smaller recurrent orbit. Section F provides further diagnostics.

7 Discussion and conclusion

PathFM combines flow matching, distillation, shared computation, and direct score refinement to generate complete future trajectories efficiently. Its strongest gains concern the quality of complete trajectories and the downstream predictions they support. The dynamical study shows a unique benefit: independently continued paths preserve variability that point forecasts can lose.

Several limitations remain. PathFM does not yet support conditioning on modalities beyond time series, such as text. For chaotic systems, all four dynamical diagnostics remain well above the held-out-truth reference (Table 2). Accurately recovering the invariant distribution remains an open challenge. A key direction for future work is to continue scaling the backbone and training data, pushing the frontier of generative time series forecasting.

AI USE STATEMENT

We used generative AI tools to support research execution, including clean and reformat dataset, implementation and analysis of experimental results. We also used these tools for literature retrieval and discovery, drafting sections of the paper, and polishing the writing for clarity and readability. We have reviewed all AI-assisted work, including code, experimental analyses, literature references, and manuscript text. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 3 describes the model and learning objectives, and Section B provides the architecture and training recipe. Sections 4 and D specify the trajectory benchmark, score estimators, and aggregation rules. Section F describes the chaotic-system experiments and their evaluation metrics. We will release the code and model weights to support reproducibility.

REFERENCES

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

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

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

Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Nicolas Chapados, and Alexandre Drouin. Tactis-2: Better, faster, simpler attentional copulas for multivariate time series. In *International conference on learning representations*, volume 2024, pp. 22897–22924, 2024.

Andreas Auer, Patrick Podest, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. Tirex: Zero-shot forecasting across long and short horizons with enhanced in-context learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 57529–57580. Curran Associates, Inc., 2025. doi: 10.52202/085713-1929. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/5356603f9c47399adfd372f77a677057-Paper-Conference.pdf.

Rongyao Cai, Yuxi Wan, Kexin Zhang, Ming Jin, Zhiqiang Ge, Daoyi Dong, Hang Yu, Yong Liu, and Qingsong Wen. The procrustean bed of time series: The optimization bias in point-wise loss functions, 2026. URL https://arxiv.org/abs/2512.18610.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International conference on learning representations*, volume 2024, pp. 57611–57640, 2024.

Ben Cohen, Emaad Khwaja, Youssef Doubli, Salahidine Lemaachi, Chris Lettieri, Charles Masson, Hugo Miccinilli, Elise Ramé, Qiqi Ren, Afshin Rostamizadeh, Jean Ogier du Terrail, Anna-Monica Toon, Kan Wang, Stephan Xie, Zongzhe Xu, Viktoriya Zhukova, David Asker, Ameet Talwalkar, and Othmane Abou-Amal. This time is different: An observability perspective on time series foundation models. *arXiv preprint arXiv:2505.14766*, 2025. URL https://arxiv.org/abs/2505.14766.

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

Valentin De Bortoli, Alexandre Galashov, J Swaroop Guntupalli, Guangyao Zhou, Kevin Murphy, Arthur Gretton, and Arnaud Doucet. Distributional diffusion models with scoring rules. *arXiv preprint arXiv:2502.02483*, 2025.

Alexandre Drouin, Étienne Marcotte, and Nicolas Chapados. Tactis: Transformer-attentional copulas for time series. In *International Conference on Machine Learning*, pp. 5447–5493. PMLR, 2022.

Christopher AT Ferro. Fair scores for ensemble forecasts. *Quarterly Journal of the Royal Meteorological Society*, 140(683):1917–1923, 2014.

William Gilpin. Chaos as an interpretable benchmark for forecasting and data-driven modelling. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=enYjtbjYJrf.

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

Jiahe Huang, Sihan Xu, Sharvaree Vadgama, and Rose Yu. Recursive flow matching, 2026. URL https://arxiv.org/abs/2605.26535.

Alexis Jacq, Guillaume Couairon, Valentin De Bortoli, Quentin Berthet, Arnaud Doucet, and Romuald Elie. Diffusion fine-tuning with rewarded moment matching distillation. *arXiv preprint arXiv:2606.30414*, 2026.

Ayush Jain and Rajat Sen. TimesFM-3: A zero-shot foundation model for multivariate forecasting. Google Research Blog, August 2026. URL https://research.google/blog/timesfm-3-a-zero-shot-foundation-model-for-multivariate-forecasting/. Published August 31, 2026; accessed September 12, 2026.

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

Marcel Kollovieh, Marten Lienen, David Lüdke, Leo Schwinn, and Stephan Günnemann. Flow matching with gaussian process priors for probabilistic time series forecasting. In *International Conference on Learning Representations*, volume 2025, pp. 95435–95457, 2025.

Jeffrey Lai, Anthony Bao, and William Gilpin. Panda: A pretrained forecast model for chaotic dynamics. In *International Conference on Learning Representations*, volume 2026, pp. 104651–104695, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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

Alvaro Perez-Diaz, James C Loach, Danielle E Toutoungi, and Lee Middleton. Foundation model forecasts: Form and function. *arXiv preprint arXiv:2510.19345*, 2025.

Romain Pic, Clément Dombry, Philippe Naveau, and Maxime Taillardat. Proper scoring rules for multivariate probabilistic forecasts based on aggregation and transformation. *Advances in Statistical Climatology, Meteorology and Oceanography*, 11(1):23–58, 2025.

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

Zhongzheng Qiao, Sheng Pan, Anni Wang, Viktoriya Zhukova, Yong Liu, Xudong Jiang, Qingsong Wen, Mingsheng Long, Ming Jin, and Chenghao Liu. It's TIME: Towards the next generation of time series forecasting benchmarks. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.12147.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL https://jmlr.org/papers/v21/20-074.html.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International conference on machine learning*, pp. 8857–8868. PMLR, 2021.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3): 1181–1191, 2020.

Michael Scheuerer and Thomas M Hamill. Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities. *Monthly Weather Review*, 143(4):1321–1334, 2015.

Neta Shaul, Chao Liu, Arash Vahdat, and Julius Berner. Parallel decoding distillation for fast image and video generation. *arXiv preprint arXiv:2607.26004*, 2026.

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

Annan Yu, Danielle Maddix, Boran Han, Xiyuan Zhang, Abdul Fatir Ansari, Oleksandr Shchur, Christos Faloutsos, Andrew Gordon Wilson, Michael W Mahoney, and Bernie Wang. Understanding the implicit biases of design choices for time series foundation models. In *International Conference on Learning Representations*, volume 2026, pp. 71998–72049, 2026.

A Related Work

Marginal forecasting. Chronos-2, TimesFM-3, and Toto 2.0 provide direct marginal quantiles while sharing information across variables (Ansari et al., 2025; Jain & Sen, 2026; Khwaja et al., 2026). Toto 1.0 instead predicts marginal Student-t mixtures and samples forecasts autoregressively across output patches (Cohen et al., 2025). Perez-Diaz et al. (2025) distinguish forecast products by their operational uses and show that marginals do not identify path-dependent event probabilities. Our diagnosis applies this distinction to forecasting benchmarks, separating missing dependence information from finite-ensemble scoring effects.

Generative trajectory forecasting. DeepAR samples temporally dependent futures autoregressively (Salinas et al., 2020), and TACTiS learns multivariate predictive distributions with attentional copulas (Drouin et al., 2022). TimeGrad introduces autoregressive diffusion for multivariate forecasting (Rasul et al., 2021). Flow matching learns generative vector fields through conditional probability paths (Lipman et al., 2022; Huang et al., 2026); TSFlow incorporates Gaussian-process priors (Kollovieh et al., 2025), and Sundial uses flow matching for foundation-model pretraining (Liu et al., 2025). Sundial also studies sample count and integration steps as inference-time calibration (Liu et al., 2025). PathFM combines a conditional backbone, connected post-training stages, and ensemble generation that reuses history computation.

Distillation and calibration. Parallel Decoding Distillation predicts multiple integration updates per network evaluation (Shaul et al., 2026). Distributional diffusion uses scoring rules to support larger generation steps (De Bortoli et al., 2025). Rewarded Moment Matching Distillation refines a distilled diffusion model with rewards and reference-based regularization; its GenCast application optimizes unbiased sample CRPS and improves dispersion (Jacq et al., 2026). We adapt PDD to joint forecasting, then differentiate fair CRPS through the one-evaluation sampler while retaining PDD supervision from the frozen flow teacher. This differs from RMMD's reward gradient through a denoising step on re-noised samples. Our study measures the quality and computational effects of this connected procedure, including whether improved marginal reliability coincides with better forecasts of trajectory-dependent quantities.

Forecast evaluation. Proper scoring rules underpin forecast comparison (Gneiting & Raftery, 2007). Fair scores address finite-ensemble effects (Ferro, 2014); variogram and transformation-based scores expose complementary multivariate properties (Scheuerer & Hamill, 2015; Pic et al., 2025). fps-bench applies these principles to future paths using fev-bench tasks (Shchur et al., 2025), with component scores reported alongside marginal accuracy. DYSTS supplies interpretable chaotic systems (Gilpin, 2021), while PANDA studies pretraining across chaotic dynamics (Lai et al., 2026). These settings let us compare short-term tracking with long-run dynamical fidelity.

B MODEL AND TRAINING

This section specifies the backbone and the training recipe for 126M and 300M models, then examines distillation, calibration, and the architecture controls that support the design. It closes with model-size comparisons of forecast accuracy and inference cost, including results on FEV, FPS, TIME, and GIFT-Eval. The main text refers to the 126M model, while the 300M model is used for scaling experiments and additional benchmarks. Model-size labels identify families; the default distilled model contains 126.3M parameters.

B.1 BACKBONE AND INPUT

Backbone configuration. The settings below describe the general-domain forecasting model; Section F gives the separate dynamical configuration. The backbone uses velocity prediction, ReLU activation, rotary temporal positions, and dynamic adaptive RMS normalization. The latter applies $(1+\alpha)\odot \mathrm{RMSNorm}(h)+\beta$, with scale and shift obtained from a shared time projection plus block-specific offsets. Modulation and velocity readouts start at zero; the learned state-entry gate starts at 0.1. Temporal and channel attention use query–key normalization; training applies weight decay 0.01. The Gaussian source scale is fixed at $\sigma = 0.75$. Both model sizes use width 768, 12 attention heads, and feed-forward width 3,072. The 126M and 300M backbones have 12 and 30 blocks, with two and ten flow-free entry blocks, respectively.

Input representation. Following the input representation of Chronos-2 (Ansari et al., 2025), each channel uses context-derived location μ_c and positive scale s_c:

$$\bar{Y}_{\ell c} = \operatorname{asinh}\left(\frac{Y_{\ell c}-\mu_c}{s_c}\right), \qquad Y_{\ell c} = \mu_c + s_c \sinh(\bar{Y}_{\ell c}). \tag{6}$$

The scale is the context root-mean-square deviation, floored at 10^{-4}. Let binary masks V and K mark valid requested coordinates and known future coordinates, respectively, and let A contain normalized known values, zero elsewhere. Only $M = V \odot (1-K)$ is transported. We write

$$\Pi_M(X) = M \odot X + K \odot A, \qquad \|B\|_M^2 := \frac{\sum_{\ell,c} M_{\ell c} B_{\ell c}^2}{\max(1, \sum_{\ell,c} M_{\ell c})}, \tag{7}$$

Here $\odot$ denotes elementwise multiplication; batch axes are included when present. The supervision mask excludes missing targets, known covariates, and final-patch padding.

Flow-time sampling. Each training row uses one scalar flow time, shared across all channels and forecast steps. For each row and draw, we sample $t = \mathrm{sigmoid}(-0.8 + 0.8\epsilon)$, $\epsilon \sim \mathcal{N}(0,1)$. The interpolated state and velocity target follow Equation (1), with this same t applied to the entire future. Four independent source/time draws share each history, with their losses averaged and normalized by the batch's transported-coordinate count. At inference, all channels advance at a common flow time.

B.2 TRAINING RECIPE

Both 126M and 300M models follow the four-stage schedule in Table 3. Transformer weights are initialized afresh while retaining the released Chronos-2 input embeddings, which remain trainable.

Stage	Updates	Backbone learning rate	Warmup	EMA decay
Flow matching	1,000,000	5×10^{-5}, constant	2,000	0.9999
Terminal decay	20,000	$5\times10^{-5}\to 0$	—	0.9999
PDD	20,000	3×10^{-5}, constant	500	0.999
PDD + fair CRPS	60,000	$3\times10^{-5}\to 0$	—	0.999

Table 3: General-corpus training schedule for both model sizes. Warmup is linear and included in the update budget; decay to zero is linear. EMA starts at the end of warmup and is preserved across continuation stages. Each complete recipe contains 1.1M optimizer updates.

Optimization and stage transitions. The reference recipe uses AdamW with $(\beta_1, \beta_2) = (0.9, 0.99)$, weight decay 0.01 on parameter matrices, global gradient-norm clipping at 1, and seed 42. An optimizer update contains 1,024 channel rows. Terminal decay continues the flow model with its optimizer and EMA state. PDD initializes the student from the terminal teacher's EMA weights, freezes that teacher, and starts a fresh optimizer. Calibration resumes the PDD student, optimizer, and EMA.

Data and task sampling. Training uses a 83M-row dataset of real-world and synthetic time series, including both univariate and multivariate data. The maximum context length is 8,192 time steps, and the maximum forecast horizon is 480 time steps. During training, we randomly sample context and forecast lengths within these limits.

Post-training objectives. PDD trains the backbone and twenty interval heads on a uniform time grid, using the block-size menu $\{1, 2, 5, 10, 20\}$, on-policy states, Euler teacher labels, and tiled residual heads. Calibration retains path-wise PDD and adds fair CRPS with coefficient one on each loss, globally normalized over scored coordinates. The CRPS term uses eight iid paths, one parallel evaluation, and the same Gaussian source scale as flow training and inference.

B.3 Distillation and evaluation budget

Figure 7 gives the full teacher–student budget comparison, extending the distillation panel in Figure 5(b) to FPS and both model sizes. Each pair is evaluated at 1, 2, 4, and 10 sequential network evaluations (NFE) with 32 stratified paths.

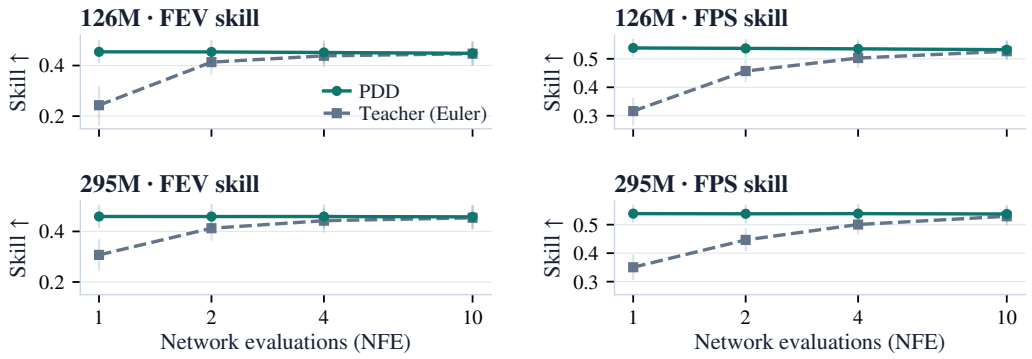

Figure 7: Expanded teacher–student comparison for Figure 5(b). Rows show teacher–student pairs from the 126M and 295M families; columns show FEV and six-component FPS skill, with higher values preferred. NFE uses a logarithmic horizontal axis. Thin vertical bars show pointwise 95% task-bootstrap intervals from 2,000 resamples.

At one evaluation, PDD avoids the teacher's sharp loss of accuracy and has higher skill scores than the ten-evaluation teacher in both pairs. Additional evaluations do not improve these estimates. This shows that the student successfully distills the teacher's knowledge into a single evaluation, and is fundamentally capped by the teacher's accuracy.

B.4 ARCHITECTURE AND TRAINING ABLATIONS

Table 4 follows the cumulative design from flow matching (R0–R5) through distillation, fair-CRPS refinement, and stratified sampling (R6–R8). The modulation, initialization, normalization, and regularization bundle in R2 improves both FEV and FPS. Restricting history attention in R3 enables shared history computation but reduces accuracy; late entry and four training draws partially recover it. PDD then improves both scores while reducing the number of evaluations from ten to one. R7 adds fair CRPS to the PDD objective, and R8 changes source sampling at the same weights.

Table 4: Cumulative architecture and post-training study. FPS always uses iid paths.

	Cumulative change	NFE	FEV ↑	FPS ↑
R0	Flow matching with clean target	10	0.4314	0.4917
R1	+ Velocity prediction	10	0.4311	0.4924
R2	+ Time modulation, zero-init head, QK norm, weight decay	10	0.4468	0.5215
R3	+ History-only attention	10	0.4356	0.5186
R4	+ Two flow-free entry blocks	10	0.4381	0.5213
R5	+ Four packed training draws	10	0.4402	0.5234
R6	+ Parallel Decoding Distillation	1	0.4471	0.5353
R7	+ PDD with fair CRPS	1	0.4488	**0.5388**
R8	+ Stratified sampling	1	**0.4575**	**0.5388**

Table 5 evaluates R0–R5 on the same held-out data, making the different prediction parameterizations comparable. The validation losses follow the benchmark pattern: R2 gives the lowest losses, R3 increases both, and R4–R5 recover part of the loss.

Table 5: Matched corpus validation and recorded training cost. Every row completes 160k architecture updates; losses use a common held-out protocol. R5's cost is reused historical training.

Row	Velocity loss ↓	Decoded-clean loss ↓	GPU-hours
R0	0.34005	0.15291	90.72
R1	0.33925	0.15261	90.72
R2	**0.33645**	**0.15122**	94.35
R3	0.34640	0.15591	95.01
R4	0.34485	0.15520	93.89
R5	0.33702	0.15149	91.54

B.5 Coverage and interval width

We compare the after-distillation 126M checkpoint with its successor after additional PDD-plus-CRPS updates on all 100 FEV tasks. Coverage increases alongside interval width for marginals, path totals, and maxima (Table 6).

Forecast object	Diagnostic	PDD	PDD + CRPS	Paired change [95% CI]
Marginal	Coverage (%)	67.00	70.63	$+3.63\,[+3.09, +4.15]$
	Interval width	3.329	3.590	$+0.262\,[+0.195, +0.338]$
Total	Coverage (%)	66.53	70.17	$+3.65\,[+2.95, +4.33]$
	Interval width	93.350	99.214	$+5.864\,[+4.017, +7.850]$
Maximum	Coverage (%)	62.81	65.75	$+2.94\,[+2.19, +3.72]$
	Interval width	4.767	5.103	$+0.336\,[+0.234, +0.443]$

Table 6: Coverage and width of empirical 10th–90th percentile intervals before and after refinement. Widths use seasonal-error scaling; totals and maxima use the same observed-coordinate mask for both models. Tasks are averaged equally after pooling valid units within each task. The 95% CIs use 10,000 paired bootstrap resamples of tasks.

B.6 Scaling and additional benchmarks

We compare the 126M and 300M models on FEV, FPS, GIFT-Eval, and TIME, as well as their cost.

B.6.1 Model size and forecast accuracy

The 126M and 300M families contain 126.3M and 296.3M learned parameters, respectively, including their distillation heads. Their training recipes are given in Section B.2.

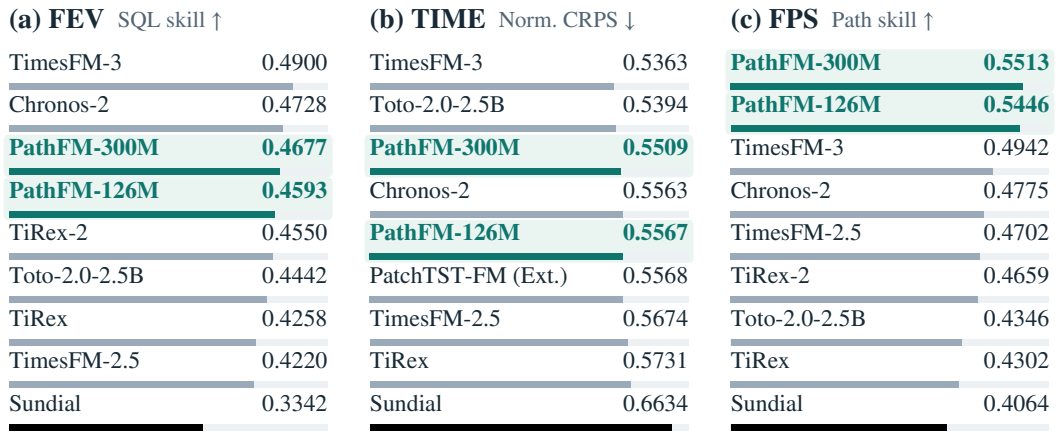

Figure 8: Scaling PathFM from the 126M to the 300M family. Panels rank nine models on FEV SQL skill (100 tasks), TIME normalized CRPS (98 tasks), and FPS path skill (94 tasks). Native samplers PathFM and Sundial are teal and black. FPS selects each quantile model's better independent or comonotone completion.

The 300M PathFM variant has higher FEV and FPS skill and lower GIFT-Eval and TIME CRPS than the 126M model used in the main benchmark comparison (Figure 8 and table 7). PathFM-300M ranks third on FEV and TIME and first on FPS among the baselines.

Table 7 adds GIFT-Eval (Aksu et al., 2024) to the benchmark suite. GIFT uses all 97 configurations without series or window caps and with cross-learning disabled. Our 126M model ranks 5th on GIFT-Eval CRPS, while the 300M model ranks 3rd.

	FEV	FPS	GIFT-Eval		TIME	
Model	Skill ↑	Skill ↑	MASE ↓	CRPS ↓	MASE ↓	CRPS ↓
TimesFM-3	**0.4900**	0.4942	—	—	**0.6398**	**0.5363**
Chronos-2	0.4728	0.4775	0.6978	0.4854	0.6620	0.5563
TiRex-2	0.4550	0.4659	0.6973	0.4781	—	—
Toto-2.0-2.5B	0.4442	0.4346	**0.6956**	**0.4759**	0.6419	0.5394
TiRex	0.4258	0.4302	0.7158	0.4885	0.6831	0.5731
TimesFM-2.5	0.4220	0.4702	0.7050	0.4903	0.6686	0.5674
Sundial	0.3342	0.4064	0.7502	0.5590	0.7576	0.6634
PathFM-126M	0.4593	0.5446	0.7073	0.4898	0.6610	0.5567
PathFM-300M	0.4677	**0.5513**	0.6994	0.4812	0.6532	0.5509

Table 7: Forecasting accuracy across four benchmarks. Darker blue indicates better performance on a linear scale within each column; bold marks the best value. Dashes indicate unavailable results.

B.6.2 Accuracy and inference cost

Figure 9 adds the 300M model to the earlier 126M timing comparison in Figure 4. Both variants use one evaluation per output block and lie on the observed accuracy–latency frontier for path forecasts. Section C.2 specifies the matched timing protocol and its limitations.

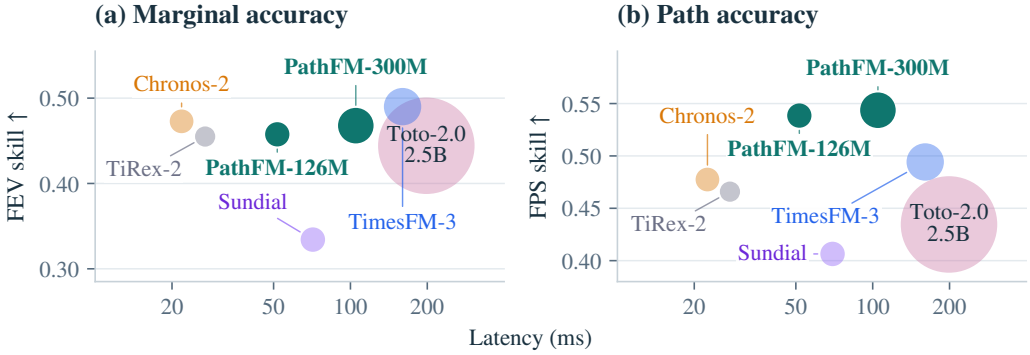

Figure 9: Accuracy, latency, and model size with the 300M PathFM variant. Circle area denotes parameter count. Both PathFM variants use 32 paths and one evaluation per output block. Latency is the median time to return nine quantiles (a) or 32 paths (b) for one query on an NVIDIA L4 at context 2,048 and horizon 288. Accuracy follows native benchmark settings.

C SAMPLING AND COMPUTATIONAL EFFICIENCY

Ensemble size determines the precision of forecast summaries, while the sampler and its execution determine their cost. We first describe sampling error and ensemble construction, then report controlled latency and memory comparisons.

C.1 FINITE-ENSEMBLE APPROXIMATION

A finite ensemble approximates a model's predictive distribution through sampled paths. This Monte Carlo approximation affects both forecast summaries and evaluation scores: estimates fluctuate across ensembles, and nonlinear score estimators can also be biased. We first separate approximation error from model error, then describe ensemble construction; Section D.1.1 derives the score correction used in evaluation.

C.1.1 FORECAST QUANTITIES AND MODEL ERROR

Condition on the available forecasting information $\mathcal{I}$ and on the fitted model. For $m \geq 1$ iid paths from $P_\theta(\cdot \mid \mathcal{I})$ and a fixed functional g of finite variance, write $\mu_\theta = \mathbb{E}_{P_\theta}[g(X) \mid \mathcal{I}]$. The sample average is unbiased for μ_θ, and independence gives

$$\widehat{\mu}_g = \frac{1}{m}\sum_{i=1}^{m} g(X^{(i)}), \qquad \mathbb{E}_Z[(\widehat{\mu}_g - \mu_\theta)^2 \mid \mathcal{I}] = \frac{\mathrm{Var}_{P_\theta}(g(X) \mid \mathcal{I})}{m}. \tag{8}$$

Thus ordinary iid Monte Carlo standard error scales as $m^{-1/2}$. This error remains for nondegenerate queries even when P_θ is the true conditional law. Direct quantile outputs avoid estimating their reported quantiles from finitely many paths, but retain model error. Efficient ensemble generation therefore matters for both inference cost and the accuracy of downstream estimates.

Let the finite value $\mu_\star$ be the corresponding expectation under the true conditional law. The sample average's squared error relative to this target decomposes as

$$\mathbb{E}_Z[(\widehat{\mu}_g - \mu_\star)^2 \mid \mathcal{I}] = (\mu_\theta - \mu_\star)^2 + \frac{\mathrm{Var}_{P_\theta}(g(X) \mid \mathcal{I})}{m}. \tag{9}$$

The cross term vanishes because $\mathbb{E}_Z[\widehat{\mu}_g - \mu_\theta \mid \mathcal{I}] = 0$. The first term is error in the model's conditional functional expectation; the second is Monte Carlo approximation error. This expression concerns estimation of a conditional expectation, not prediction error against one random future outcome, which also contains outcome variability.

For an event A with model probability $p_\theta = P_\theta(A \mid \mathcal{I})$, taking $g(X) = \mathbf{1}\{X \in A\}$ gives sampling variance $p_\theta(1-p_\theta)/m$. For $0 < p_\theta < 1$, the relative standard error is $\sqrt{(1-p_\theta)/(mp_\theta)}$, making rare-event probability estimates demanding at small ensemble sizes. Increasing m reduces this sampling term but does not remove model error. Using conditional marginals or analytical integration, when available for a particular query, need not incur the same sampling error. Dependent or stratified draws require a different variance analysis. The $m^{-1/2}$ rate describes the approximation error of the iid sample average at a finite sample budget.

C.1.2 ENSEMBLE CONSTRUCTION AND STRATIFICATION

At inference, we approximate the forecast distribution with an ensemble of m complete future trajectories, from which we estimate quantiles and other forecast summaries. Ensemble size controls the number of paths, independently of the one- or two-evaluation budget per path. With few draws, these summaries can be sensitive to sampling variability. For more even source coverage at fixed m, each coordinate uses the midpoints of m equiprobable Gaussian bins, independently permuted across paths:

$$u_r = \frac{r - \frac{1}{2}}{m}, \qquad Z^j_{\ell c} = \sigma\Phi^{-1}(u_{\pi_{\ell c}(j)}), \quad r, j \in \{1, \ldots, m\}, \tag{10}$$

Here Φ is the standard-normal CDF and each $\pi_{\ell c}$ is an independent permutation. This assigns every rank once per coordinate without tying a path's ranks across times or channels. The fair estimator's iid unbiasedness guarantee does not apply to these stratified ensembles. Calibration and unbiased fair-score evaluation use iid Gaussian draws.

C.2 Inference latency

Table 8 compares complete serving calls on an NVIDIA L4 with context 2,048, horizon 288, and batches of one or sixteen histories. We measure end-to-end time, with CUDA synchronization around each call. We report medians over 35 measured calls in seven blocks, following warmup and with randomized execution order. Models retain their native behavior. The marginal interface returns nine quantiles. The path interface returns 32 paths or independent completion of the quantile outputs.

Table 8: End-to-end latency on an NVIDIA L4 at context 2,048 and horizon 288.

		Nine quantiles (ms)		32 paths (ms)	
Model	Params. (M)	Batch 1	Batch 16	Batch 1	Batch 16
PathFM, one evaluation	126.3	51.7	215.3	51.6	215.2
PathFM, two evaluations	126.3	96.4	403.2	96.4	403.3
Chronos-2	119.5	21.8	66.6	22.5	72.9
TimesFM-3	330.7	160.0	508.2	160.6	515.1
Toto-2.0-2.5B	2,454.3	198.3	1,482.6	199.0	1,489.5
TiRex	35.3	170.4	180.9	170.9	187.1
TiRex-2	82.5	27.0	44.0	27.6	50.5
Sundial	128.3	71.2	1,118.3	69.6	1,085.1

With one evaluation, PathFM generates 32 paths in 51.6 ms for one history and 215.2 ms for sixteen. Batching sixteen histories reduces the average time per history to 13.5 ms. At both batch sizes, PathFM with one evaluation is faster than Sundial and TimesFM-3. Chronos-2 and TiRex-2 offer lower latency, while PathFM achieves higher FPS skill (Figure 4).

C.2.1 Inference speedup breakdown

Figure 10 isolates execution choices in a separate study on A100. Packing paths reduces teacher latency from 2,663.8 to 827.1 ms per batch; the packed two-evaluation student takes 201.9 ms. Adding deeper history K/V caching changes teacher latency only to 801.4 ms and is disabled for the student. The combined student speedup therefore comes from packing and distillation. Packing also reduces peak PyTorch allocated memory from 3.22 to 1.58 GiB.

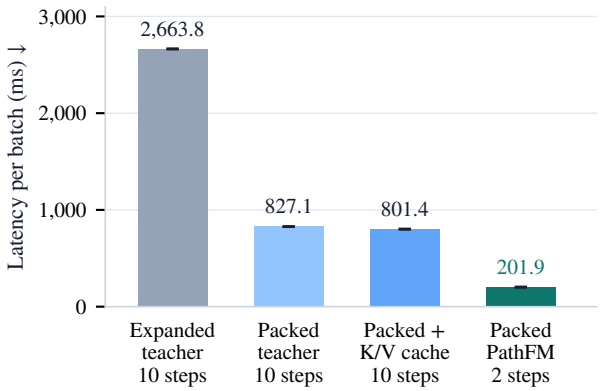

Figure 10: Inference speedup breakdown on an A100 at batch 16, context 2,048, horizon 480, and 32 stratified paths. The teacher uses ten Euler evaluations; the student uses two. All configurations use float32 weights with bfloat16 autocast.

D EVALUATION METRICS

We specify the score estimators, aggregation rules, and finite-ensemble effects needed to interpret the forecasting results. The chaotic-system protocols and diagnostics are given separately in Section F.

D.1 SCORE CONSTRUCTION

D.1.1 FINITE-ENSEMBLE ESTIMATORS

For a fixed outcome y and predictive law P, CRPS and energy score have the common form

$$S_\rho(P, y) = \mathbb{E}\rho(X, y) - \tfrac{1}{2}\mathbb{E}\rho(X, X'), \qquad X, X' \overset{\text{iid}}{\sim} P, \tag{11}$$

using absolute and Euclidean distance, respectively (Gneiting & Raftery, 2007). For $m \geq 2$ iid paths and integrable distances, the off-diagonal estimator

$$\widehat{S}_{\rho,\text{fair}} = \frac{1}{m}\sum_{i=1}^{m}\rho(x^{(i)}, y) - \frac{1}{2m(m-1)}\sum_{i\neq j}\rho(x^{(i)}, x^{(j)}) \tag{12}$$

is unbiased for $S_\rho(P, y)$. Using the empirical distribution instead replaces $m(m-1)$ by m^2 and introduces the shift

$$\mathbb{E}\widehat{S}_{\rho,\text{emp}} - S_\rho(P, y) = \frac{1}{2m}\mathbb{E}\rho(X, X'). \tag{13}$$

The shift follows from the m zero diagonal distances and penalizes dispersion even at a fixed ensemble size (Ferro, 2014). We use Equation (12) for all CRPS and energy components, but its unbiasedness guarantee applies only to iid ensembles, not the requested-stratified main comparison. The variogram component retains a squared sample mean (Equation (15)); fixing $m = 32$ does not remove its model-dependent bias.

D.1.2 TEMPORAL PATH SCORES

For item n and channel c, define the seasonal-error scale from history h and seasonal period p:

$$a_{nc} = \frac{1}{|\mathcal{H}_{nc}|}\sum_{t\in\mathcal{H}_{nc}}|h_{ntc} - h_{n,t-p,c}|, \qquad \mathcal{H}_{nc} = \{t : h_{ntc}, h_{n,t-p,c} \text{ observed}\}. \tag{14}$$

Except for first passage, the temporal scores use paths divided by finite positive a_{nc}. Table 9 specifies their distances in Equation (12); the variogram uses its separate estimator below.

Score	Distance or transformation
Energy (ES)	$\rho(x, y) = \|x - y\|_2$
Differenced ES (DES)	$\rho(x, y) = \|D(x) - D(y)\|_2$, where $D(x) = (x_1, x_2 - x_1, \ldots, x_L - x_{L-1})$
Variogram (VS)	Squared errors in expected temporal contrasts, Equation (15)
Sum CRPS	$\rho(x, y) = \lvert\sum_t x_t - \sum_t y_t\rvert$
Max CRPS	$\rho(x, y) = \lvert\max_t x_t - \max_t y_t\rvert$
First-passage CRPS	$\rho(u, v) = \lvert T_\tau(u) - T_\tau(v)\rvert$ in original units, Equation (16)

Table 9: The six temporal FPS components. Each component scores one target channel's temporal path before aggregation across channels and tasks.

Level, sum, and maximum scores use the same observed-coordinate mask for every forecast path. DES retains the initial value when observed and increments only when both adjacent observations are available.

Variogram. For the nonempty set of observed time pairs $\mathcal{P} = \{(s, t) : s < t,\ y_s, y_t \text{ observed}\}$, we use

$$\widehat{\text{VS}}_{1/2} = \frac{1}{|\mathcal{P}|}\sum_{(s,t)\in\mathcal{P}}\left(|y_s - y_t|^{1/2} - \frac{1}{m}\sum_{i=1}^{m}|x_s^{(i)} - x_t^{(i)}|^{1/2}\right)^2. \tag{15}$$

All pairs receive equal weight (Scheuerer & Hamill, 2015).

First passage. The threshold τ is the maximum of the last available context segment of length at most L, in original units. For a path u,

$$T_\tau(u) = \min\big(\{t \in \{1, \dots, L\} : u_t > \tau\} \cup \{L + 1\}\big). \tag{16}$$

This score requires a fully observed future and finite threshold, uses forecast steps as units, and treats no crossing as $L + 1$. It therefore evaluates a censored event time. The sum, maximum, and first-passage scores identify their respective functional distributions rather than the full path law.

D.1.3 Cross-variable variogram score

The cross-variable variogram compares channel pairs at the same forecast time. Channels are standardized by their mean and standard deviation over at most 8,192 context observations; insufficient history or negligible variance excludes a channel. Using shared observed-coordinate masks, the estimator in Equation (15) averages over valid times and channel pairs, items, origins, and finally tasks with an arithmetic mean. We report this absolute score.

D.2 Aggregation rules

Marginal benchmarks. We retain the official within-task reductions for FEV, GIFT-Eval, and TIME (Shchur et al., 2025; Aksu et al., 2024; Qiao et al., 2026). FEV uses nine quantiles at $\{0.1, 0.2, \dots, 0.9\}$. SQL scales errors by a_{nc} before averaging. Official all-task SQL also retains the benchmark's missing-result and leakage-imputation rules. GIFT-Eval and TIME report equally weighted geometric means of task or configuration losses relative to Seasonal Naive, without ratio clipping. Their quantile-grid CRPS differs from fair sample CRPS.

fps-bench. FPS compares each model with Seasonal Naive across six path metrics on the same 94 tasks with valid scores for every model and metric. For each metric, we average valid items within each channel, then channels and forecast origins, to obtain one loss per task. Let $r_{d,k}$ be the model's loss divided by Seasonal Naive's loss for task d and metric k, clipped to $[0.01, 100]$. The aggregate score is

$$\mathrm{FPS} = 1 - \mathrm{GM}_{d,k}(r_{d,k}), \tag{17}$$

where GM denotes the geometric mean with equal weight for each task and metric. A score of zero matches Seasonal Naive; higher is better. Individual component skills use the same rule for one metric at a time.

E FORECAST ACCURACY AND DEPENDENCE

These experiments examine which aspects of a forecast improve when the model generates complete paths. We report temporal component scores, evaluate cross-variable relationships, and test explicit dependence completions of quantile forecasts. Metric definitions and aggregation rules are collected in Section D.

E.1 TEMPORAL FORECAST ACCURACY

Figure 11 compares the six FPS components for PathFM and the three strongest baselines (selecting the best-performing completion for each). Table 10 reports the full comparison. For quantile models, independent completion usually gives better temporal scores than comonotone completion, but the latter can improve sum skill. PathFM achieves the highest skill on all six components, with a 9.96% paired FPS loss reduction against TimesFM-3.

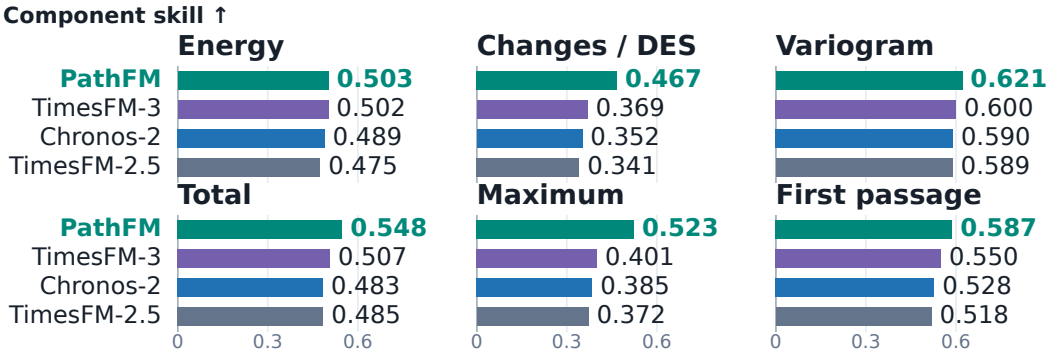

Figure 11: Component skills for PathFM and the three strongest baselines on fps-bench. Bars show skill relative to Seasonal Naive; higher is better.

Table 10: Temporal-path skill on the same 94 tasks; higher is better.

Forecast	Completion	ES	DES	VS	Sum	Max	FPT	FPS skill
TimesFM-3	Independent	0.5019	0.3688	0.5998	0.5067	0.4007	0.5500	0.4942
TimesFM-3	Comonotone	0.4843	0.3676	0.4894	0.5211	0.3534	0.5534	0.4666
Chronos-2	Independent	0.4894	0.3518	0.5901	0.4834	0.3849	0.5277	0.4775
Chronos-2	Comonotone	0.4706	0.3621	0.4818	0.5029	0.3436	0.5230	0.4515
TimesFM-2.5	Independent	0.4750	0.3412	0.5893	0.4852	0.3723	0.5184	0.4702
TiRex-2	Independent	0.4715	0.3455	0.5790	0.4698	0.3718	0.5214	0.4659
Toto-2.0-2.5B	Independent	0.4480	0.3002	0.5333	0.4381	0.3173	0.5274	0.4346
TiRex	Independent	0.4442	0.3067	0.5419	0.4363	0.3279	0.4883	0.4302
Sundial	Native	0.3838	0.3676	0.5072	0.4346	0.3060	0.4195	0.4064
Toto-1.0	Native	0.3735	0.0875	0.3583	0.4454	0.0712	0.4933	0.3236
PathFM	Native	**0.5025**	**0.4674**	**0.6213**	**0.5484**	**0.5231**	**0.5874**	**0.5446**

E.2 CROSS-VARIABLE VARIOGRAM SCORES

Table 11 compares absolute cross-variable variogram scores on all 35 multi-target fev-bench tasks, retaining all 625 forecast origins. The score compares pairs of target variables at the same forecast time, using the standardization and reduction in Section D.1.3.

The Gaussian copula uses shrinkage estimates of normal-score residual-rank correlation and a shared AR(1) temporal factor, fitted on four forecast windows ending before the first test origin. Six short-history tasks use shorter calibration horizons. The nine marginal quantiles are interpolated linearly, with constant tails outside probabilities 0.1–0.9.

Task-specific adaptation. The fitted Gaussian copula estimates dependence parameters from historical forecast–outcome pairs on each benchmark task. The forecasting weights remain frozen, but the copula introduces task-specific parameter fitting. We report these adapted references in a separate table group from PathFM and the fixed independent and comonotone completions, which require no additional task-specific parameter fitting at evaluation time.

Forecast	Copula	VS ↓
No task-specific fitting		
Chronos-2	Independent	0.1491
Chronos-2	Comonotonic	0.1648
TimesFM-3	Independent	0.1446
TimesFM-3	Comonotonic	0.1609
Toto-2.0-2.5B	Independent	0.1560
Toto-2.0-2.5B	Comonotonic	0.1741
PathFM	Native	**0.1443**
Task-calibrated references		
Chronos-2	Fitted Gaussian	0.1469
TimesFM-3	Fitted Gaussian	**0.1420**
Toto-2.0-2.5B	Fitted Gaussian	0.1533

Table 11: Contemporaneous cross-variable variogram scores on 35 multi-target tasks. Entries are arithmetic task means; lower is better, with the lowest value in each protocol group bold. Fitted Gaussian references use task-specific calibration.

E.3 Fitted Gaussian temporal copulas

We test whether fitting a Gaussian temporal copula to historical forecast errors improves the path forecasts of frozen quantile models. Building on the independent and comonotonic completions in Section E.1, we estimate temporal dependence from target-task history while retaining each baseline's marginal quantile forecasts.

Gaussian copula fitting. For each baseline and item, four forecast windows ending before the first test origin supply standardized residuals, which we rank-transform to normal scores. We estimate a lag-one coefficient within these windows, share it across target channels, clip it to $[-0.95, 0.95]$, and fix it before testing. Items with insufficient information use a task-pooled estimate. All eligible items are retained; 27 tasks use shorter calibration horizons without changing their test windows. The fitted coefficient defines a stationary latent AR(1) process used to generate temporally dependent ranks for the baseline's interpolated marginal quantiles, with constant tails outside probabilities 0.1–0.9. This parameterization restricts the fitted temporal dependence to AR(1).

Forecast	Copula	FPS ↑
No task-specific fitting		
Chronos-2	Independent	0.4775
Chronos-2	Comonotonic	0.4515
TimesFM-3	Independent	0.4942
TimesFM-3	Comonotonic	0.4666
Toto-2.0-2.5B	Independent	0.4346
Toto-2.0-2.5B	Comonotonic	0.4234
PathFM	Native	**0.5446**
Task-calibrated references		
Chronos-2	Fitted Gaussian AR(1)	0.5150
TimesFM-3	Fitted Gaussian AR(1)	**0.5311**
Toto-2.0-2.5B	Fitted Gaussian AR(1)	0.4846

Table 12: Effect of fitting a Gaussian temporal copula on FPS over 94 common fev-bench tasks. Each baseline uses the same marginal forecasts across copula completions. Fitted Gaussian rows use task-specific calibration. Bold marks the best value within each protocol group.

Fitting the Gaussian copula improves FPS from 0.4775 to 0.5150 for Chronos-2, from 0.4942 to 0.5311 for TimesFM-3, and from 0.4346 to 0.4846 for Toto-2.0-2.5B, relative to independent completion. However, these completions impose a stationary Gaussian AR(1) dependence structure, and PathFM achieves higher FPS without task-specific copula fitting.

F Chaotic systems

We examine how model size, sampling, and fine-tuning affect point accuracy and long-run dynamics.

F.1 Setup and forecasting protocols

Models and training. The quantitative comparisons use the 75M and 126M PathFM families with native 256-step horizons, trained for 1M flow-matching updates followed by 20k terminal-decay updates. We train the 75M variant to approximately match PANDA's parameter count (74.2M versus 71.6M). The 126M backbone has 12 transformer blocks and feedforward width 3,072; the 75M model uses 11 blocks and width 768. Both initialize their backbones afresh while retaining trainable input-patch embeddings. Main results use ten-step Euler sampling, source standard deviation 0.86, and seed 42.

Point forecasts. We evaluate six 512-context windows from each of 9,347 trajectories using sMAPE, as in PANDA (Lai et al., 2026). We average sMAPE within 480 source groups and report their median. PathFM feeds back the coordinatewise median of 16 stratified samples per 256-step block, averaging the middle two order statistics. Chronos-1 uses 16 samples and median feedback every 128 steps, chaining two native 64-step calls within each sample; Chronos-2 and PANDA continue point forecasts in 128-step blocks.

F.2 Model size and sampler choice

Table 13 compares PathFM with 72M-parameter PANDA. The smaller PathFM retains lower errors at all four forecasting horizons and on all four dynamical diagnostics, showing that the gains persist at similar model size. We obtain these gains with our general forecasting architecture and flow-matching objective, without adding architectural components or losses tailored to chaotic systems. PANDA uses specialized dynamics embeddings with random polynomial and Fourier features motivated by Koopman-operator approximations (Lai et al., 2026).

Table 13: Model size at ten Euler evaluations per block. Both PathFM sizes follow the protocol in Section F.1; lower is better, with the best value in bold.

	sMAPE				Dynamics			
Model	$H = 64$	$H = 128$	$H = 256$	$H = 512$	W_1	PSD H^2	CD err.	Seg. analog
PANDA	14.96	20.62	29.57	40.19	0.678	0.181	0.437	10.03
PathFM-75M	13.19	18.63	26.55	34.44	0.214	**0.046**	0.166	4.52
PathFM-126M	**12.59**	**18.08**	**25.37**	**33.39**	**0.202**	0.047	**0.166**	**4.42**

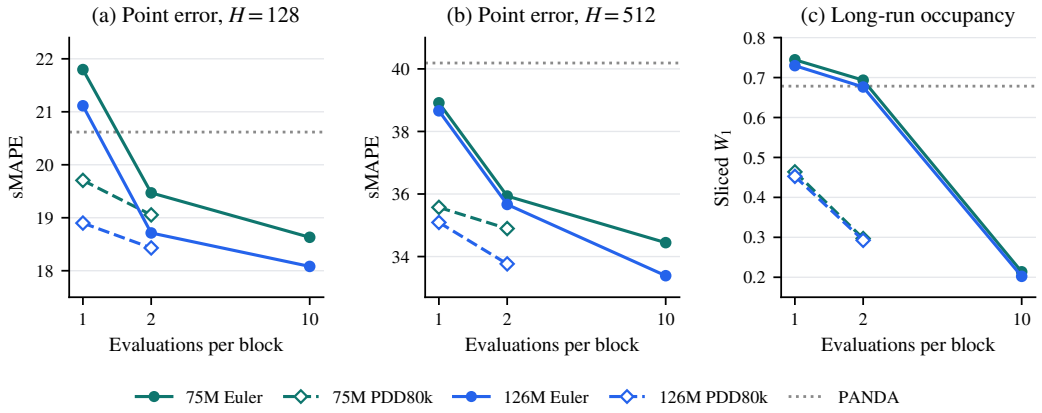

Figure 12: Accuracy versus sampler budget for both model sizes. Point errors summarize 480 source groups; W_1 summarizes 135 systems and 4,096-step sampled continuations. The dotted horizontal line marks PANDA; lower is better.

The sampler comparison evaluates each terminal-decay teacher and its student after 80k PDD updates (Figure 12 and table 14). At one or two evaluations, PDD improves point accuracy over Euler; ten-step Euler remains best within each size at every horizon. For 126M long continuations, two-evaluation PDD reduces W_1 from 0.676 to 0.292 relative to two-step Euler, approaching ten-step Euler's 0.202.

Table 14: Point-forecast sMAPE for the Euler teacher and its student after 80k PDD updates. NFE counts evaluations per 256-step block and sample; bold marks the lowest error within each size.

Model	Sampler	NFE	$H = 64$	$H = 128$	$H = 256$	$H = 512$
PathFM-75M	Euler	1	15.76	21.80	30.22	38.92
	Euler	2	13.63	19.47	28.02	35.93
	Euler	10	**13.19**	**18.63**	**26.55**	**34.44**
	PDD80k	1	14.01	19.71	27.41	35.57
	PDD80k	2	13.50	19.05	27.00	34.90
PathFM-126M	Euler	1	15.22	21.11	29.73	38.66
	Euler	2	13.00	18.71	27.51	35.67
	Euler	10	**12.59**	**18.08**	**25.37**	**33.39**
	PDD80k	1	13.24	18.90	26.71	35.09
	PDD80k	2	12.70	18.43	25.98	33.77

F.3 Full continuations

Figure 13 extends the qualitative comparison in Figure 6 to all 4,096 steps and adds Rössler. The displayed PathFM-75M paths use ten Euler evaluations per block. Sampling sustains Lorenz's two-lobe motion; the Rössler sample remains oscillatory but has distorted occupancy. The Rössler example is a failure case for all evaluated models, highlighting the need for future work on preserving long-run dynamics.

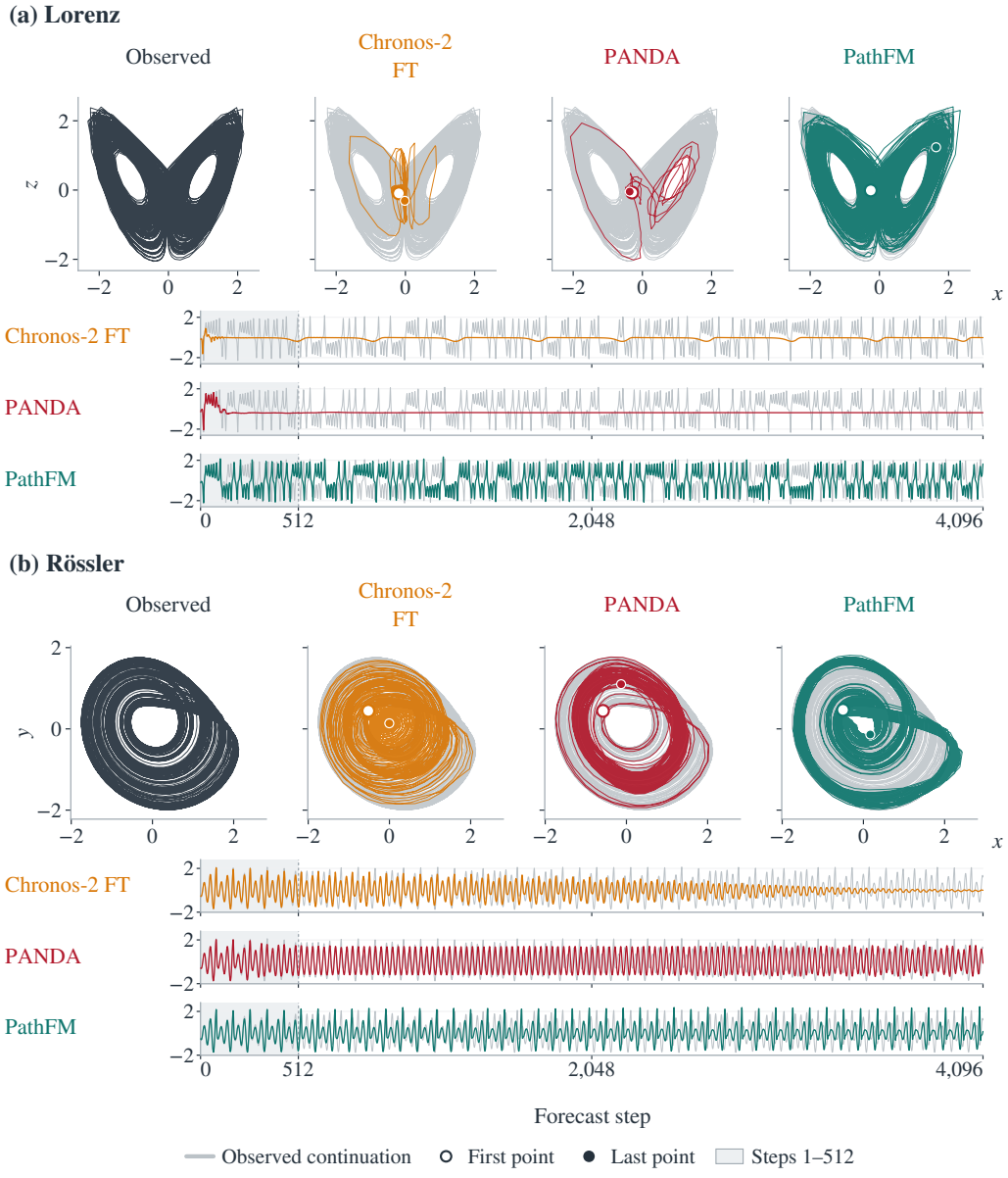

Figure 13: Full 4,096-step Lorenz and Rössler continuations. All panels use the same initial conditions. Gray curves show the observed continuation; circles mark the first and last forecast states, and shading marks the first 512 steps.

F.4 METRIC DEFINITIONS

Point error. With context channel mean μ_c and population standard deviation σ_c, let $\widetilde{x}_{tc} = (x_{tc} - \mu_c)/\max(\sigma_c, 10^{-10})$ and define $\widetilde{y}$ identically. We use PANDA's factor-100 convention (Lai et al., 2026):

$$\mathrm{sMAPE}_h = \frac{100}{hC} \sum_{t=1}^{h} \sum_{c=1}^{C} \frac{|\widetilde{x}_{tc} - \widetilde{y}_{tc}|}{|\widetilde{x}_{tc}| + |\widetilde{y}_{tc}| + 10^{-10}}. \tag{18}$$

This scores the blockwise median forecast; individual sampled-path errors are not averaged into the point score.

State-distribution distance. Let $\mathcal{R}$ be the reference state cloud and $\mathcal{S}$ the post-burn-in rollout states, standardized using reference channel means and standard deviations. For $K = 64$ seeded unit directions, $\widehat{F}_{\mathcal{R},k}$ and $\widehat{F}_{\mathcal{S},k}$ are the empirical CDFs of the projected states:

$$\widehat{\mathrm{SW}}_1(\mathcal{R}, \mathcal{S}) = \frac{1}{K} \sum_{k=1}^{K} \int_0^1 \left| \widehat{F}^{-1}_{\mathcal{R},k}(u) - \widehat{F}^{-1}_{\mathcal{S},k}(u) \right| du. \tag{19}$$

The reference cloud retains at most 20,000 states; this calculation subsamples each cloud to at most 10,000. It measures distance to an empirical state distribution, without establishing an invariant measure of the learned process.

Spectral distance. Spectral and segment diagnostics standardize channels using trajectories 0–7. Welch spectra use 512-point periodic Hann windows, 50% overlap, unit sampling rate, and constant detrending. Each spectrum is normalized to unit mass; $p_{x,c}$ is the forecast spectrum and q_c averages the six normalized reference spectra. The channel-mean squared Hellinger distance is

$$H^2_{\mathrm{spec}} = \frac{1}{2C} \sum_{c=1}^{C} \sum_{f} \left[\sqrt{p_{x,c}(f)} - \sqrt{q_c(f)} \right]^2. \tag{20}$$

It lies in $[0, 1]$ and compares spectral shape independently of total power.

Correlation dimension. We estimate $\widehat{d}_{\mathrm{corr}}$ from log–log slopes of empirical pair-distance probabilities, excluding pairs within a 30-step Theiler window. The estimator uses at most 4,096 states and 200,000 positive pair distances, taking the median slope over probability ranges $[0.005, 0.05]$, $[0.01, 0.10]$, and $[0.02, 0.20]$, with 20 geometrically spaced probabilities per range. Against the dataset's stored pooled estimate $\widehat{d}_{\mathrm{pool}}$, we report

$$E_{\mathrm{CD}} = \left| \widehat{d}_{\mathrm{corr}} - \widehat{d}_{\mathrm{pool}} \right|. \tag{21}$$

The pooled target uses all ten original trajectories, including the test trajectory, and its own standardization; it is a descriptive geometric reference rather than an independent held-out estimate.

Segment analog. Flatten 64-step multivariate segments into vectors in the shared standardized coordinates, without recentering each segment. Let $\mathcal{U}_x$, $\mathcal{U}_{\mathrm{ref}}$, and $\mathcal{U}_{\mathrm{cal}}$ denote generated, reference, and calibration segments, and $d(u, \mathcal{U}) = \min_{v \in \mathcal{U}} \|u - v\|_2$. The score compares their nearest-reference distances:

$$A_{\mathrm{seg}} = \frac{\mathrm{median}_{u \in \mathcal{U}_x}\, d(u, \mathcal{U}_{\mathrm{ref}})}{\max\{\mathrm{median}_{u \in \mathcal{U}_{\mathrm{cal}}}\, d(u, \mathcal{U}_{\mathrm{ref}}), 10^{-12}\}}. \tag{22}$$

Generated and calibration segments use stride 32; reference segments use stride eight, retaining at most 20,000. The calibration normalization gives a reference level of one; larger values indicate less similar local dynamics.