CHIRP-MODAL GAUSSIAN MIXTURES FOR JOINT FORECASTING AT IRREGULAR TIMES

Anonymous authors
Paper under double-blind review

ABSTRACT

Probabilistic forecasting of irregular time series demands coherent uncertainty across future times. Accurate pointwise intervals alone may neglect the temporal dependence that governs plausible trajectories. We propose chirp-modal Gaussian mixtures, which integrate damped oscillations with an explicit temporal covariance. Within each component, frequency and decay dynamically shift within the forecast window, enabling both the mean and the alignment of uncertainty to adapt over time. A commuting dynamical structure affords analytic means and transitions, while positive numerical variance propagation enables efficient joint sampling at irregular queries. We generate these components directly from history (CMD-X) or via latent reverse diffusion (CMD-D), and learn their temporal covariance with full-grid likelihoods. Across four forecasting benchmarks, direct chirp mixtures yield lower mean energy and variogram scores relative to fixed-pole mixtures at comparable trainable parameter counts. At the primary configuration, iterative generation further improves joint scores, and same-checkpoint comparisons validate keeping mixture structure beyond a single Gaussian. Our approach integrates adaptable temporal dependence with a choice of direct or iterative generation.

1 INTRODUCTION

An air-quality forecast might output the correct exceedance probability at each hour while failing to evaluate the probability of a sustained pollution episode. An episode is a property of a path: marginal intervals fail to indicate whether elevated values co-occur at adjacent queries. A reliable forecast needs to reflect both pointwise uncertainty and dependence across times (Gneiting et al., 2007; Scheuerer & Hamill, 2015). The same historical data might lead to quick pollution dissipation or persistence. These alternative regimes motivate a mixture of distinct futures, while correlated fluctuations around either regime require temporal dependence within each component. A collection of plausible means alone leaves that second source of uncertainty unaddressed.

The dynamics of a regime can adapt during the forecast window. Oscillations may accelerate or decelerate, shifting the alignment of future deviations; changing decay alters how long adisturbance persists. These within-window changes demand time-varying modal dynamics. Irregular sampling introduces a distinct problem: unequal elapsed times determine how much phase and damping accumulate between observations or queries. Equal query counts rarely correspond to equal physical durations (Weerakody et al., 2021; Schirmer et al., 2022). Missing observations and channels additionally limit the signals available to infer the current state and future regime (Che et al., 2018; Shukla & Marlin, 2021; Yalavarthi et al., 2024). A forecast must consequently condition on incomplete histories and compute its transitions at the actual requested times.

Conditional flows, diffusion models, and continuous-time state models already capture joint forecasts (Yalavarthi et al., 2025; 2026; Rasul et al., 2021; Tashiro et al., 2021; Schirmer et al., 2022). This study investigates whether a compact, history-conditioned component family can adapt its modal dynamics within a window while maintaining tractable joint uncertainty. LLapDiff (You et al., 2026) offers a foundational framework through stable latent modal means at irregular queries. Its frequency and decay are time-invariant in physical time within each denoiser evaluation. Collections of these fixed poles can model complex signals; enabling the poles themselves to evolve offers an alternative mechanism to represent changing phase and persistence.

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.

We propose ***chirp-modal Gaussian mixtures***. Each component integrates evolving frequency and decay with a stochastic modal state. A commuting dynamical structure yields analytic means and transitions, while isotropic noise simplifies temporal covariance propagation to scalar energy updates. The phase controls temporal alignment; decay and accumulated noise control persistence and variance. Positive numerical energy increments construct a valid Gaussian law for correlated path sampling. Mixing these laws models alternative regimes, with one component identity maintained throughout each path. The construction unifies chirplet representations of changing frequency (Mann & Haykin, 1995) with explicit nonstationary covariance (Remes et al., 2017).

The component family and its generator are independent design axes. CMD-X predicts the component parameters directly from history and learned codes as a mixture density network (Bishop, 1994). CMD-D yields a component from each latent reverse-diffusion trajectory using deterministic DDIM updates (Song et al., 2020). Both learn temporal covariance through full-grid likelihoods. The direct generator enables a single-pass forecast; the iterative generator progressively shapes components through successive noisy-state updates.

Our primary contributions can be summarized as follows:

- **Time-varying modal Gaussian family.** We propose a time-varying modal Gaussian family characterized by analytic transitions and positive semidefinite numerical covariance.
- **Direct and iterative mixture generators.** We develop direct and iterative mixture generators with explicit likelihood objectives. At the primary configuration, CMD-D delivers better mean joint scores, while CMD-X offers accelerated sampling.
- **Evidence for the proposed distributional structure.** Across four empirical benchmarks, CMD-X decreases mean energy score by 4.0–27.9% compared with closely capacity-matched fixed-pole mixtures. Fixed-pole comparisons and pole-field ablations validate evolving dynamics, while same-checkpoint interventions and a synthetic regime experiment elucidate the roles of within-component covariance and persistent component identity.

2 METHODOLOGY

2.1 OVERVIEW: A JOINT FORECAST AT IRREGULAR TIMES

Forecasting setup. Let $x_j \in \mathbb{R}^{d_x}$ be an observation at time t_j and $o_j \in \{0,1\}^{d_x}$ its channel-wise observation mask. A history summarizer derives $E = S_\phi(\mathcal{H})$ from $\mathcal{H} = \{(t_j, x_j, o_j) : t_j \le t_i\}$. Our model predicts at the h requested times $Q = (t_1^q, \dots, t_h^q)$. Offsets $t_r = t_r^q - t_i \ge 0$ maintain the physical lead gap. A frozen VAE encodes future targets as $z_r \in \mathbb{R}^{d_z}$ and decodes sampled paths pointwise; the history summarizer also remains frozen during predictor training. Scores are in observation space (Appendix D).

One component family, two generators. We formulate a plausible future regime by a joint Gaussian $P_{\Theta,Q} = \mathcal{N}(m_{\Theta,Q}, K_{\Theta,Q})$ over the requested latent vectors. Its parameters Θ consist of frequency and decay coefficients, modal loadings ℓ_{kd}, process-noise rates q_k, initial variances p_k^0, and output scale α. Dynamics and variance parameters are identical across latent coordinates d; the loadings are coordinate-specific. We aggregate M such components into

$$P_Q = \frac{1}{M} \sum_{j=1}^{M} P_{\Theta^{(j)},Q}. \tag{1}$$

We propose two generators for this family (Figure 1). In CMD-X, a shared head outputs $\Theta^{(j)} = G_\theta(E, e_j)$ from learned component codes e_j, generating all components in one pass. In CMD-D, a reverse-diffusion map predicts $\Theta^{(j)} = F_\theta(E, Q, u_j)$ from terminal noise $u_j \sim \mathcal{N}(0, I_{hd_z})$. Here F_θ defines the complete reverse procedure on Q, each reverse trajectory supplies one component.

Temporal uncertainty at two levels. A path maintains one component identity. Its kernel characterizes correlated residual variation; different components represent alternative futures. Within a component, latent coordinates are independent. Shared identity and the per-time decoder induce dependence across observed channels.

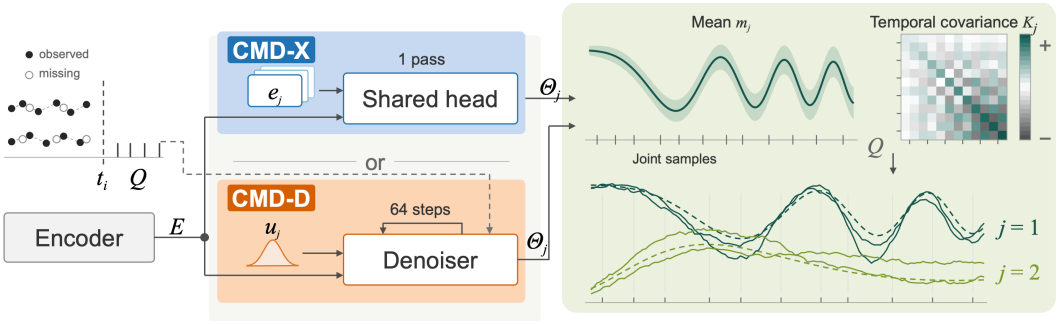


Figure 1: **Schematic overview of the two generators.** CMD-X produces all components in one pass; CMD-D uses 64 reverse steps per component. Both predict a mean m_j and temporal covariance K_j on the query grid Q. Each joint sample retains one component identity for the whole path. In the lower panel, dashed curves denote component means and solid curves denote joint samples.

2.2 Chirped dynamics with analytic transitions

We use K damped oscillatory modes on a forecast interval $[0, H]$ containing the requested offsets. The duration H differs from the query count h and the harmonic time scale L (Appendix B). The modal frequencies and decay rates dynamically shift within this interval. For mode k, $\rho_k(t)$ is the instantaneous decay rate and $\omega_k(t)$ is the angular frequency. We express their cumulative decay and phase as

$$R_k(t) = \int_0^t \rho_k(v)\, dv, \qquad \Omega_k(t) = \int_0^t \omega_k(v)\, dv. \tag{2}$$

Each field is a constant pole plus a harmonic perturbation:

$$\rho_k(t) = \rho_k^{\mathrm{f}} + \sum_{m=1}^{4} a_{\rho,km}^2 \phi_m(t), \qquad \omega_k(t) = \omega_k^{\mathrm{f}} + \sum_{m=1}^{4} a_{\omega,km}^2 \phi_m(t). \tag{3}$$

The signed cosine features ϕ_m have analytic integrals. Coefficient rescaling bounds the fields and keeps decay positive; zero coefficients recover fixed poles (Appendix B).

Let I_2 be the planar identity, $J = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$, and $\mathrm{Rot}(\theta)$ a planar rotation through angle θ. We use the dynamics and transition

$$A_k(t) = -\rho_k(t) I_2 + \omega_k(t) J, \qquad \Phi_k(t,s) = e^{-[R_k(t) - R_k(s)]} \mathrm{Rot}(\Omega_k(t) - \Omega_k(s)). \tag{4}$$

Remark 1 (Commuting dynamics)**.** Each $A_k(t)$ is a linear combination of the fixed commuting matrices I_2 and J, thus the dynamics commute across time. Equation (4) is therefore satisfied without numerical ODE integration. Lemma 1 in Appendix A.2 proves this identity and the transition composition property.

2.3 An explicit temporal distribution around each mean

Independent state copies across latent coordinates give separate temporal covariance blocks; isotropic modal noise allows for scalar variance propagation. With process-noise rate $q_k > 0$ and initial variance $p_k^0 > 0$, we employ two-dimensional states ξ_{kd}, mutually independent standard Brownian motions W_{kd}, and independent initial noises $\eta_{kd} \sim \mathcal{N}(0, I_2)$:

$$d\xi_{kd} = A_k \xi_{kd}\, dt + \sqrt{q_k} I_2\, dW_{kd}, \qquad \xi_{kd}(0) = e_{\mathrm{init}} + \sqrt{p_k^0}\eta_{kd}, \qquad z_d(t) = \alpha \sum_k \ell_{kd}^\top \xi_{kd}(t). \tag{5}$$

Here $e_{\mathrm{init}} = (1, 0)^\top$, $\ell_{kd} = (c_{kd}, b_{kd})^\top$ contains the modal loadings, and α is an output scale. We obtain the analytic mean

$$m_d(t) = \alpha \sum_{k=1}^{K} e^{-R_k(t)} \left[c_{kd} \cos \Omega_k(t) + b_{kd} \sin \Omega_k(t) \right]. \tag{6}$$

To describe uncertainty, we define the *accumulated modal energy*

$$G_k(t) = p_k^0 + q_k \int_0^t e^{2R_k(v)}\, dv. \tag{7}$$

It aggregates initial uncertainty with noise accumulated over time, before applying the damping back to the state. This single scalar function specifies each mode's temporal covariance.

Proposition 1 (Temporal kernel)**.** *Condition on a component's parameters. For absolutely continuous R_k, Ω_k, G_k on $[0, H]$, with G_k nondecreasing and $G_k(0) = p_k^0 \geq 0$, the kernel*

$$\mathcal{K}_{dd'}(t,s) = \mathbf{1}_{\{d=d'\}}\alpha^2 \sum_k \|\ell_{kd}\|^2 e^{-R_k(t)-R_k(s)} G_k(\min\{t,s\}) \cos(\Omega_k(t) - \Omega_k(s)) \tag{8}$$

is positive semidefinite. Together with (6)*, it defines a Gaussian component; using the energy in* (7) *recovers the state model* (5)*.*

Appendix A derives the kernel and a shared-state extension.

2.4 How chirps shape dependence and enable joint sampling

For one active scalar mode with positive variance and $s \leq t$, the kernel gives

$$\mathrm{Corr}(z_d(t), z_d(s)) = \sqrt{G_k(s)/G_k(t)}\cos(\Omega_k(t) - \Omega_k(s))\,. \tag{9}$$

Phase drift shifts temporal alignment while preserving modal marginal variance. A chirp makes the phase-alignment factor $\cos(\Omega_k(t) - \Omega_k(s))$ depend on location within the window as well as lag. At constant frequency this factor is $\cos(\omega_k(t-s))$; the energy ratio in (9) can still depend on both times.

Positive variance propagation. From the last-observation origin $t_0 = 0$, sort the query offsets and define $\Delta t_r = t_r - t_{r-1}$ and $\Delta R_{kr} = R_k(t_r) - R_k(t_{r-1})$. For modal marginal variance $\Lambda_k(t)$ and innovation variance $V_k(t,s)$, hats denote query-grid approximations:

$$\begin{aligned} \Lambda_k(t) = e^{-2R_k(t)}G_k(t), \qquad V_k(t,s) = e^{-2R_k(t)}[G_k(t) - G_k(s)] \geq 0, \\ \widehat{\Lambda}_{kr} = e^{-2\Delta R_{kr}}\widehat{\Lambda}_{k,r-1} + \widehat{V}_{kr}, \qquad \widehat{\Lambda}_{k0} = p_k^0, \\ \widehat{V}_{kr} = q_k \Delta t_r \psi(\Delta R_{kr}) \geq 0, \qquad \psi(x) = \frac{1-e^{-2x}}{2x}, \quad \psi(0) = \lim_{x\to 0}\psi(x) = 1. \end{aligned} \tag{10}$$

The numerical covariance relies on the kernel in (8) with $G_k(t_r)$ replaced by $\widehat{G}_{kr} = e^{2R_k(t_r)}\widehat{\Lambda}_{kr}$. The initial Gaussian state and independent Gaussian innovations with variances $\widehat{V}_{kr} \geq 0$ form a positive-semidefinite covariance under the analytic transitions. The variance integral is approximated; the resulting numerical Gaussian law yields the training covariance and enables exact sequential joint sampling. Appendix B outlines the proof, stable evaluation near zero, and the fixed-anchor construction.

Remark 2 (Positive semidefiniteness and tractability)**.** Positive-definite supervised blocks support the full-grid likelihood in Section 2.5; singular blocks necessitate regularization. Modal sampling costs $O(Khd_z)$ per component, whereas dense blockwise likelihood factorization costs $O(d_z h^3)$.

2.5 Learning the full-grid likelihood

We learn both means and temporal covariances with the *full-grid Gaussian negative log-likelihood (NLL)* of encoded future targets. The set $\mathcal{O}_t$ comprises query times with at least one observed target entry; all latent coordinates contribute at those times. Missing observed channels do not mask out individual latent coordinates. With $n_{\mathcal{O}} = d_z|\mathcal{O}_t|$, the component log density is

$$\ell_Q(\Theta; z) = \sum_{d=1}^{d_z} \log \mathcal{N}(z_{\mathcal{O}_t,d}; m_{\mathcal{O}_t,d}, K_{d,\mathcal{O}_t}). \tag{11}$$

Here $m_{\mathcal{O}_t,d} = (m_d(t_r))_{r\in\mathcal{O}_t}$ and $K_{d,\mathcal{O}_t} = (\mathcal{K}_{dd}(t_r,t_s))_{r,s\in\mathcal{O}_t}$ restrict the mean and the dth temporal block of $K_{\Theta,Q}$; dependence on the fixed component and grid is suppressed locally. Their

dimensions are $|\mathcal{O}_t|$ and $|\mathcal{O}_t| \times |\mathcal{O}_t|$. Throughout training and sampling, K is constructed by the positive variance recurrence.

Direct mixture learning. We train CMD-X on clean latent futures using the full-grid mixture NLL,

$$\mathcal{L}_X = -\mathbb{E}\left[\frac{1}{n_{\mathcal{O}}}\log\left\{\frac{1}{M_{\text{train}}}\sum_{j=1}^{M_{\text{train}}}\exp \ell_Q(G_\theta(E, e_j); z)\right\}\right]. \tag{12}$$

This directly fits the forecast mixture, maintaining one component identity across times and coordinates. Fixed-pole controls share this objective and head design.

Iterative denoising learning. We train CMD-D to predict a clean future from a noisy latent window. At a sampled diffusion level τ, let $\bar{\alpha}_\tau$ be the cumulative signal-retention factor and $\epsilon \sim \mathcal{N}(0, I_{hd_z})$. We form $z_\tau = \sqrt{\bar{\alpha}_\tau} z + \sqrt{1-\bar{\alpha}_\tau}\epsilon$ and minimize

$$\mathcal{L}_D = -\mathbb{E}_{z,E,Q,\tau,\epsilon}\left[\frac{w_\gamma(\tau)}{n_{\mathcal{O}}}\ell_Q(\Theta_\theta(z_\tau, \tau, E); z)\right], \tag{13}$$

Here Θ_θ estimates a clean-window Gaussian conditional and w_γ weights diffusion levels. Unlike $\mathcal{L}_X$, $\mathcal{L}_D$ fits a denoising conditional; inference derives the forecast mixture using 64 deterministic DDIM updates of predicted clean means (Song et al., 2020). Each terminal-noise trajectory returns one component for subsequent correlated path sampling and per-time decoding (Algorithms 1–2).

3 Experiments

Setup. We evaluate NOAA-UK (meteorology, seven stations, $h = 168$), USHCN (climate records, 1,114 stations, $h = 12$), BMS-air (air quality, 12 sites, $h = 168$), and PhysioNet ($h = 12$, LLapDiff setting) (You et al., 2026; Menne & Williams Jr, 2009; De Brouwer et al., 2019; Zhang et al., 2017). The environmental datasets retain observation gaps and missing channels; USHCN also uses irregular forecast queries. Their chronological 70/10/20 splits precede windowing, with training-set normalization (Appendix D). CMD-X/D, fixed-pole (LTI) heads, and Chirp-MSE use the shared frozen latent pipeline. QuITE++, TFMixer, SDE-VI, MOSES, ProFITi, CRU, NCDSSM, and PatchTST model directly from raw observations with their own trained representations, and use their own probabilistic heads. Appendix C outlines the remaining controls and LLapDiff. Final evaluations employ ten seeds, a 200-epoch final-stage limit, and model-specific best-validation-loss selection; baseline search uses 8–12 seed-0 trials (Appendix K).

Energy score (ES) assesses the joint forecast, variogram score of order 0.5 (VS) highlights dependence, and CRPS evaluates marginal accuracy (Gneiting & Raftery, 2007; Scheuerer & Hamill, 2015). VS uses unnormalized inverse-time-separation weights with separation floored by each window's median consecutive gap (Appendix D). We present means over ten seeds (0–9), using 25 observation-space scoring paths, and evaluate interval coverage alongside width. Appendix Tables 4–5 specify the models; Tables 6–13 give complete metrics and available seed standard deviations.

3.1 Does the chirp component family improve forecasting?

Drifting poles improve the direct mixture. The comparison in Table 1 investigates pole drift while holding the direct-mixture design fixed: CMD-X and the LTI mixtures share conditioning, learned-code generation, uniform weights, the whole-window mixture objective, and query-grid covariance evaluation. Increasing the LTI modal count yields trainable parameter counts within 0.04% of CMD-X on NOAA-UK/USHCN and 0.20% on BMS-air. Relative to these larger LTI heads, CMD-X decreases mean ES by **4.0–27.9%** and VS by **3.8–42.0%**, and improves CRPS on all four benchmarks. Equal-K controls give the same ordering. These gains support within-window pole drift. MSE-trained mean-synthesizer comparisons on USHCN also favor chirps (Appendix Table 15).

Which pole fields require drift? At basis width four, varying both pole fields yields lower mean ES and VS than varying either alone (Table 2); this remains true at all three tested widths on NOAA-UK and USHCN. Increasing the harmonic width continues to benefit the both-field direct head on both datasets (Figure 7, Appendix L).

Table 1: **Component family first; iteration second.** Means at $M = 25$ with 25 scoring paths; lower scores are better (best results in bold, second-best results underlined). Both chirp predictors use $K = 64$; LTI (matched) matches CMD-X's capacity ($K = 79$ for NOAA-UK/USHCN/PhysioNet, $K = 74$ for BMS-air). Seed standard deviations are reported in the appendix.

	PhysioNet				USHCN			
Model	ES	VS	CRPS	PIT-ECE	ES	VS	CRPS	PIT-ECE
LTI ($K = 64$)	1.6428	4.7021	0.3750	0.0175	2.1628	0.5845	0.3918	0.2092
LTI (matched)	1.6323	4.7020	0.3720	**0.0160**	2.1629	0.5864	0.3913	0.2078
CMD-X	1.5674	4.5241	**0.3422**	0.0204	1.5596	0.4635	0.2607	**0.0528**
CMD-D	**1.4763**	**4.3916**	0.3453	0.0381	**1.5151**	**0.4541**	**0.2575**	0.0795

	NOAA-UK				BMS-air			
Model	ES	VS	CRPS	PIT-ECE	ES	VS	CRPS	PIT-ECE
LTI ($K = 64$)	16.345	915.35	0.5681	0.1077	14.259	861.08	0.5964	0.0889
LTI (matched)	16.268	894.77	0.5661	0.1119	14.402	756.30	0.6045	0.0863
CMD-X	13.771	518.61	0.4593	0.1159	11.613	659.12	**0.3823**	0.0337
CMD-D	**13.608**	**438.80**	**0.4503**	**0.0943**	**11.246**	**469.64**	0.3846	**0.0182**

Table 2: **Frequency and decay drift at fixed basis width.** CMD-X with $K = 64$, $M_\phi = 4$, and $M = 25$; 25 scoring paths. Test mean $\pm$ standard deviation over ten seeds; lower is better (best results in bold, second-best results underlined). Reported parameter counts are 250,912 for both fields and 232,224 for either alone (Appendix L).

	NOAA-UK		USHCN	
Varying fields	ES	VS	ES	VS
Both	**13.771** $\pm$ 0.168	**518.61** $\pm$ 20.98	**1.5596** $\pm$ 0.0053	**0.4635** $\pm$ 0.0039
Frequency only	14.719 $\pm$ 0.295	642.4 $\pm$ 37.0	1.5802 $\pm$ 0.0089	0.4834 $\pm$ 0.0034
Decay only	15.365 $\pm$ 0.368	661.3 $\pm$ 73.8	1.5861 $\pm$ 0.0057	0.4824 $\pm$ 0.0039

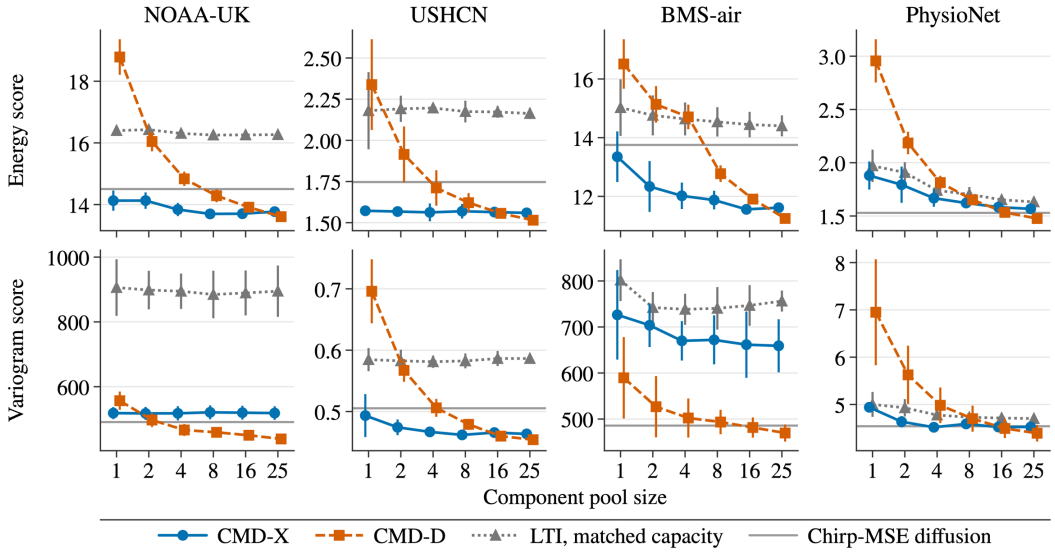


Figure 2: **Component-pool evaluation across four datasets.** Points show mean ES and VS with one standard deviation over ten training seeds, using 25 scoring paths at every setting. Direct heads retain prefixes of a 25-code checkpoint; CMD-D generates the requested pool. LTI uses 79/79/74/79 modes in dataset order. Horizontal lines show the Chirp-MSE reference means. Appendix D specifies the sample allocation.

3.2 GENERATOR QUALITY, COMPONENT POOLS, AND COMPUTATION

Direct versus iterative generation. At basis width four and $M = 25$, CMD-D reduces mean ES by 1.2–5.8% and VS by 2.0–28.7% in comparison to CMD-X across all four benchmarks. CMD-X achieves lower CRPS on BMS-air and PhysioNet. Both predictors contain roughly 0.25M trainable forecasting parameters (Appendix K). Appendix Tables 6–9 detail variability for all operating points.

Component-pool evaluation. Direct heads generate all 25 trained components in one pass and retain learned-code prefixes of size $M_{\text{pool}} \in \{1, 2, 4, 8, 16, 25\}$, while CMD-D generates the requested pool from exchangeable terminal draws. Each pool supplies 25 scoring paths using the allocation in Appendix D.

Increasing the pool lowers CMD-D's mean ES and VS on all four datasets in Figure 2, while direct curves are not strictly monotonic. At pool size eight, CMD-X has lower mean ES on all four benchmarks and lower VS on USHCN and PhysioNet (Appendix Tables 6–9). At 25, CMD-D leads on both joint scores, with CMD-X within 6.2% in mean ES.

Measured generation cost. On the three environmental datasets, the generation times for one window and 25 decoded paths on an RTX A6000 are 19–20 ms for CMD-X and 128–151 ms for batched CMD-D. The corresponding latency ratios are 6.7–7.6× (Appendix Figure 6; protocol in Appendix I).

3.3 WHICH DISTRIBUTIONAL STRUCTURE MATTERS?

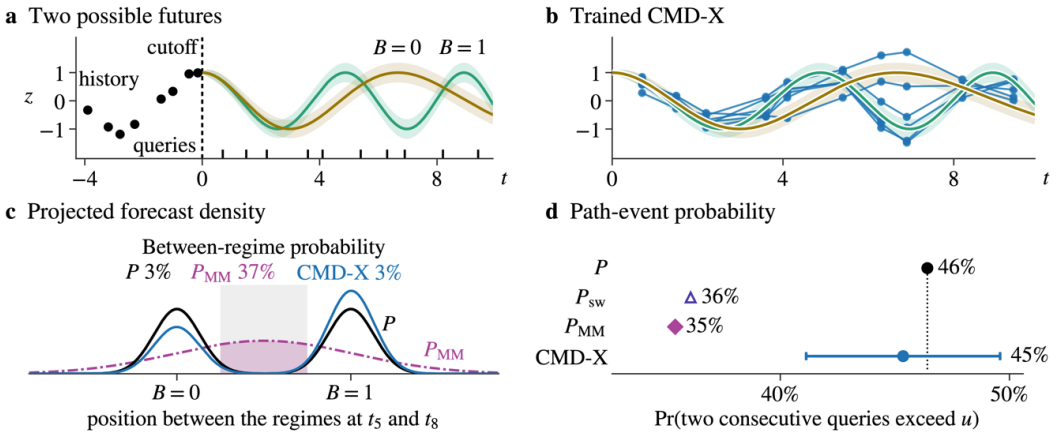


Figure 3: **Learning coherent futures from samples.** (a) Oracle P retains one chirped regime per path; bands show ± 1 residual standard deviation. (b) Six joint CMD-X draws. (c) Projected densities at queries 5 and 8; shading marks the region between regimes. P_{MM} matches P's full mean and covariance. (d) Adjacent-query exceedance probabilities; P_{sw} redraws the regime at each query, preserving one-time marginals. Panels (b–c) show seed 0; CMD-X in (d) shows mean $\pm$ standard deviation across ten training seeds. Appendix H gives the protocol.

Learning distinct, coherent futures. We trained a 25-component CMD-X on 20,000 unlabeled paths from two chirped regimes with correlated residuals, using a scalar latent, fixed context, and full-grid mixture NLL. The learned mixture recovers the regimes and their low-density separation (Figure 3b–c); a Gaussian matching the oracle's complete mean and covariance fills in that separation. Across ten training seeds, CMD-X's excess joint NLL relative to the oracle is 0.003 nats/query; Gaussian collapse raises it to 0.100. Redrawing the oracle regime at each query maintains one-time marginals but changes exceedance probabilities at adjacent queries (Figure 3d). CMD-X's event estimates are close on average but vary with learned regime shares (Appendix H). Neither complete first and second moments nor correct marginals determine the joint forecast.

Three same-checkpoint interventions. For each training seed, we apply three interventions to the same CMD-D checkpoint using the same 25 terminal draws: **(i) Block-diagonal** K_j, **(ii) Moment-matched Gaussian**, and **(iii) Component means as paths**. Block-diagonal K_j maintains per-time

distributions and shared component identity while removing within-component temporal covariance (Proposition 3). For the moment-matched Gaussian, we use

$$\bar{m} = \frac{1}{M}\sum_{j=1}^{M} m_j, \qquad \bar{K} = \frac{1}{M}\sum_{j=1}^{M} K_j + \frac{1}{M}\sum_{j=1}^{M}(m_j - \bar{m})(m_j - \bar{m})^\top. \tag{14}$$

These expressions align with the finite mixture's full latent mean and covariance, including temporal and cross-coordinate terms. For component means as paths, we decode each latent component mean as a complete path, retaining variation between components while removing within-component residual variation. This assesses whether diversity among component means alone suffices to represent forecast uncertainty. All three interventions worsen mean ES and VS on all four datasets (Figure 4), validating the combination of mixture structure and within-component temporal variation.

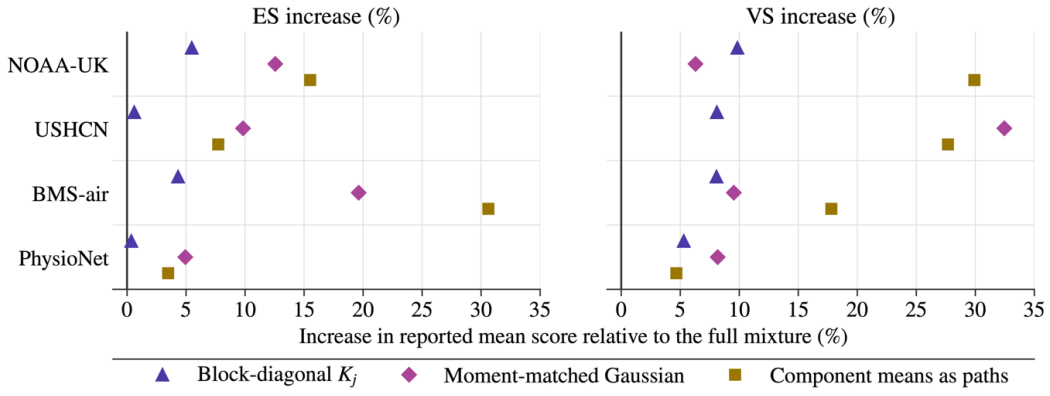


Figure 4: **Distributional interventions.** Percentage increases in reported mean ES and VS relative to the full CMD-D mixture. Raw scores and seed standard deviations: Appendix E.

3.4 Comparison with contemporary joint forecasters

External forecasting systems. CMD-D achieves lower mean ES, VS, and CRPS than every listed external baseline on all four benchmarks (Table 3). The strongest external model by mean ES varies across datasets: TFMixer + DG on NOAA-UK, SDE-VI on USHCN, QuITE++ + DG on BMS-air, and LLapDiff on PhysioNet. Relative to LLapDiff, CMD-D reduces mean ES by 9.9–22.7%. CMD-X also improves ES and CRPS over QuITE++, TFMixer, and SDE-VI across all four datasets.

Table 3: **Broader comparison.** Mean ES / VS / CRPS with 25 observation-space paths; lower scores are better (best results in bold, second-best results underlined). DG indicates the evaluated diagonal-Gaussian adaptation. Chirp-MSE diffusion shares the latent backbone and chirp-modal mean but relies on MSE training and 25 reverse trajectories. Appendix Tables 10–13 give complete metrics and seed standard deviations.

Method	NOAA-UK	USHCN	BMS-air	PhysioNet
CMD-D(25)	**13.608 / 438.8 / 0.450**	**1.515 / 0.454 / 0.258**	**11.246 / 469.6** / 0.385	**1.476 / 4.392** / 0.345
CMD-X(25)	13.771 / 518.6 / 0.459	1.560 / 0.464 / 0.261	11.613 / 659.1 / **0.382**	1.567 / 4.524 / **0.342**
Chirp-MSE diffusion	14.504 / 490.9 / 0.502	1.747 / 0.505 / 0.277	13.752 / 485.7 / 0.403	1.529 / 4.539 / 0.362
LLapDiff	16.276 / 811.0 / 0.549	1.959 / 0.554 / 0.387	13.211 / 710.4 / 0.594	1.639 / 4.840 / 0.376
QuITE++ + DG	19.300 / 918.1 / 0.664	1.686 / 3.928 / 0.324	12.922 / 1213.3 / 0.528	2.291 / 6.277 / 0.383
SDE-VI	20.207 / 630.9 / 0.709	1.656 / 2.522 / 0.299	13.986 / 627.3 / 0.588	2.304 / 5.350 / 0.386
TFMixer + DG	14.409 / 1165.2 / 0.549	1.942 / 5.476 / 0.615	13.093 / 795.1 / 0.511	2.409 / 10.133 / 0.407
MOSES	22.895 / 719.8 / 0.821	1.820 / 2.661 / 0.385	17.252 / 592.8 / 0.551	2.253 / 10.293 / 0.413
ProFITi	16.172 / 902.9 / 0.512	1.701 / 2.169 / 0.364	15.112 / 864.8 / 0.589	1.787 / 6.153 / 0.376
CRU	21.300 / 770.0 / 0.745	1.997 / 3.956 / 0.481	21.621 / 724.2 / 0.732	2.559 / 11.829 / 0.434
NCDSSM	24.395 / 820.8 / 0.854	2.247 / 2.480 / 0.327	17.973 / 613.0 / 0.579	2.442 / 7.712 / 0.392
PatchTST + Gaussian	21.972 / 587.3 / 0.658	2.377 / 4.198 / 0.405	16.321 / 557.3 / 0.620	2.391 / 9.715 / 0.457

Explicit components versus a path ensemble. CMD-D and Chirp-MSE diffusion share the latent representation and chirp-modal mean architecture. At 25 components or paths, each requires $64 \times 25 = 1{,}600$ trajectory-level denoiser evaluations, which can be batched into 64 network invocations. Compared with Chirp-MSE's MSE-trained ensemble, CMD-D's denoising likelihood and explicit component covariance decrease mean ES by 3.5–18.2%, VS by 3.3–10.6%, and CRPS by 4.6–10.4%. CMD-X achieves better CRPS on all four benchmarks and ES on the three environmental datasets. Complete calibration and interval diagnostics are detailed in Appendix E.

4 Related work

Continuous-time modal prediction. Continuous-time latent-state models progress between irregular observations (Rubanova et al., 2019; De Brouwer et al., 2019; Schirmer et al., 2022; Ansari et al., 2023). LLapDiff parameterizes stable modal means based on history and diffusion state (You et al., 2026); CMD integrates within-window pole drift and stochastic temporal covariance. The direct fixed-pole controls adopt CMD-X's likelihood objective (Appendix J).

Physics-inspired structure. Physical structure can constrain architectures, demonstrated in Hamiltonian neural networks (Greydanus et al., 2019), or regularize optimization via governing-equation residuals, as in physics-informed neural networks and diffusion models (Raissi et al., 2019; Bastek et al., 2025). Our rotation, positive decay, and stochastic forcing impose a dynamical bias without enforcing known physical equations on the observations.

Gaussian processes and state-space models. Linear SDEs formally relate Gaussian processes to Kalman computation (Hartikainen & Särkkä, 2010; Särkkä & Solin, 2019). Spectral mixtures characterize oscillatory covariance (Wilson & Adams, 2013); nonstationary spectral kernels adapt frequency with input (Remes et al., 2017). CMD unifies history-conditioned pole fields, commuting transitions, and isotropic noise to derive covariance through scalar energies.

Mixtures and joint irregular forecasting. Direct prediction of mixture parameters aligns with the mixture-density-network construction (Bishop, 1994); the novel contribution here is the component's chirp-modal temporal law. ProFITi models joint conditional distributions at irregular time-channel queries through conditional normalizing flows (Yalavarthi et al., 2025). MOSES couples Gaussian-process source distributions with separable flows and history-conditioned mixing to achieve marginalization consistency (Yalavarthi et al., 2026). We instead study chirp-modal covariance within Gaussian mixtures and compare direct and iterative generators of that family.

Recent irregular and structural forecasting. QuITE aggregates irregular observations with learned query tokens; QuITE++ builds a hierarchical forecasting encoder and query decoder around that idea (Lim, 2026). TFMixer fuses a learnable non-uniform Fourier transform with query-based local temporal mixing (Qiu et al., 2026). We evaluate QuITE++ and TFMixer with diagonal-Gaussian output heads. SDE-VI leverages SDE-induced continuous-discrete variational inference (Yuan et al., 2026), a distinct route to continuous-time probabilistic modeling.

PULSE disentangles phase-anchored structure from residual forecasting and generates future phase evolution (Liu et al., 2026). CMD instead positions evolving modes inside an explicit stochastic covariance and Gaussian mixture law. Diff-MN utilizes diffusion to jointly generate time series and sample-specific mixture-of-experts Neural Controlled Differential Equation dynamics parameters (Zhang et al., 2026). It parallels the concept of generated dynamics, but studies continuous generation with a different dynamical family and protocol.

5 Conclusion

Chirp-modal Gaussian mixtures bridge evolving modal dynamics to explicit temporal uncertainty. Commuting transitions keep the construction tractable, while stochastic covariance and shared component identity model fluctuations within regimes and alternatives between them. Direct chirp mixtures improve on closely capacity-matched fixed-pole mixtures across four benchmarks. At the primary configuration, iterative generation further improves joint scores; direct generation enables faster paths and competitive marginal accuracy. These results support an adaptable component family with a practical choice of generator conditioned on forecast quality and computational requirements. Limitations and future directions are discussed in Appendix M.

AI USE STATEMENT

We adopted a responsible and targeted approach to the use of generative AI tools during the preparation of this manuscript, utilizing them strictly when necessary to improve writing efficiency, code structuring, and overall readability. We maintained full human oversight throughout the project. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Core Research Integrity. We have not used generative AI tools for generating synthetic datasets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, assisting with translation, cleaning and reformatting dataset, supporting qualitative and thematic data analysis, interpreting results.

Writing and Proof Presentation. We used generative AI tools for assisting in the writing of proofs. All proofs were derived by the authors. We used AI tools exclusively to review our LaTeX code and suggest phrasing or formatting adjustments to enhance the clarity, flow, and readability of our mathematical arguments. All AI-suggested text and equation formatting was carefully reviewed and verified for absolute correctness by the authors prior to inclusion.

Retrieval and Related Work. We utilized generative AI in conjunction with search engines to accelerate literature discovery and assist in parsing complex prior work. However, all intellectual insights, comparisons, and syntheses presented in the paper were generated entirely by the authors. AI tools were used solely to polish the readability of our synthesized thoughts, and all resulting text was rigorously fact-checked for accuracy.

Code Implementation. We used generative AI to assist in implementing our methods, specifically to draft boilerplate PyTorch code. The underlying architecture, algorithmic logic, and input/output pipelines of all AI-assisted code were manually designed, extensively tested, and definitively verified for correctness by the authors.

REPRODUCIBILITY STATEMENT

Appendices A–C provide derivations, numerics, model specifications, and algorithms; Appendices D–E document evaluation and results. Final proposed-model, LTI, and external-baseline runs use seeds 0–9, a 200-epoch final-stage limit, and best-validation-loss checkpoint selection. Appendix H specifies the synthetic experiments and same-checkpoint controls; Appendix K records the training and selection protocols.

REFERENCES

Abdul Fatir Ansari, Alvin Heng, Andre Lim, and Harold Soh. Neural continuous-discrete state space models for irregularly-sampled time series. In *International Conference on Machine Learning*, pp. 926–951. PMLR, 2023.

Jan-Hendrik Bastek, WaiChing Sun, and Dennis Kochmann. Physics-informed diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 3360–3385, 2025.

Christopher M Bishop. Mixture density networks. 1994.

Jared Boyer, T Konstantin Rusch, and Daniela Rus. Learning to dissipate energy in oscillatory state-space models. *arXiv preprint arXiv:2505.12171*, 2025.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085, 2018.

Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. Gru-ode-bayes: Continuous modeling of sporadically-observed time series. *Advances in neural information processing systems*, 32, 2019.

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

Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 69(2): 243–268, 2007.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7407–7417. IEEE, 2023.

Jouni Hartikainen and Simo Särkkä. Kalman filtering and smoothing solutions to temporal gaussian process regression models. In *2010 IEEE international workshop on machine learning for signal processing*, pp. 379–384. IEEE, 2010.

JungHoon Lim. Quite: Query-based irregular time series embedding. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Yangyou Liu, Zezhi Shao, Xinyu Chen, Hu Chen, Fei Wang, and Yuankai Wu. Pulse: Generative phase evolution for non-stationary time series forecasting. In *Forty-third International Conference on Machine Learning*, 2026.

Steve Mann and Simon Haykin. The chirplet transform: Physical considerations. *IEEE Transactions on Signal Processing*, 43(11):2745–2761, 1995.

Matthew J Menne and Claude N Williams Jr. Homogenization of temperature series via pairwise comparisons. *Journal of Climate*, 22(7):1700–1717, 2009.

Xiangfei Qiu, Kangjia Yan, Xvyuan Liu, Xingjian Wu, and Jilin Hu. Bridging time and frequency: A joint modeling framework for irregular multivariate time series forecasting. *arXiv preprint arXiv:2602.00582*, 2026.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

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.

Sami Remes, Markus Heinonen, and Samuel Kaski. Non-stationary spectral kernels. *Advances in neural information processing systems*, 30, 2017.

Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

Wilson J Rugh. *Linear system theory*. Prentice-Hall, Inc., 1996.

T Konstantin Rusch and Daniela Rus. Oscillatory state-space models. In *International Conference on Learning Representations*, volume 2025, pp. 22303–22322, 2025.

Simo Särkkä and Arno Solin. Applied stochastic differential equations. *Institute of mathematical statistics textbooks*, 2019.

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.

Mona Schirmer, Mazin Eltayeb, Stefan Lessmann, and Maja Rudolph. Modeling irregular time series with continuous recurrent units. In *International conference on machine learning*, pp. 19388–19405. PMLR, 2022.

Satya Narayan Shukla and Benjamin M Marlin. Multi-time attention networks for irregularly sampled time series. In ... *International Conference on Learning Representations*, volume 2021, pp. 14897, 2021.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in neural information processing systems*, 34:24804–24816, 2021.

Philip B Weerakody, Kok Wai Wong, Guanjin Wang, and Wendell Ela. A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing*, 441:161–178, 2021.

Andrew Wilson and Ryan Adams. Gaussian process kernels for pattern discovery and extrapolation. In *International conference on machine learning*, pp. 1067–1075. PMLR, 2013.

Vijaya Krishna Yalavarthi, Kiran Madhusudhanan, Randolf Scholz, Nourhan Ahmed, Johannes Burchert, Shayan Jawed, Stefan Born, and Lars Schmidt-Thieme. Grafiti: Graphs for forecasting irregularly sampled time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16255–16263, 2024.

Vijaya Krishna Yalavarthi, Randolf Scholz, Stefan Born, and Lars Schmidt-Thieme. Probabilistic forecasting of irregularly sampled time series with missing values via conditional normalizing flows. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21877–21885, 2025.

Vijaya Krishna Yalavarthi, Randolf Scholz, Christian Klötergens, Kiran Madhusudhanan, Stefan Born, and Lars Schmidt-Thieme. Reliable probabilistic forecasting of irregular time series through marginalization-consistent flows. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 85403–85427, 2026.

Zinuo You, Jin Zheng, and John Cartlidge. Latent laplace diffusion for irregular multivariate time series. *arXiv preprint arXiv:2605.19805*, 2026.

Zexin Yuan, Qinliang Su, and Junxi Xiao. Generative modeling of irregular time series via SDE-induced continuous-discrete variational inference. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=IXdjkxDXI0.

Shuyi Zhang, Bin Guo, Anlan Dong, Jing He, Ziping Xu, and Song Xi Chen. Cautionary tales on air-quality improvement in beijing. *Proceedings. Mathematical, physical, and engineering sciences*, 473(2205):20170457, 2017.

Xu Zhang, Junwei Deng, Chang Xu, Hao Li, and Jiang Bian. Diff-mn: Diffusion parameterized moe-ncde for continuous time series generation with irregular observations. *arXiv preprint arXiv:2601.13534*, 2026.

A COMPONENT DERIVATIONS

This appendix derives the transition, mean, and covariance sequentially required for component instantiation. We initially show why the time-varying dynamics formally yield an ordinary matrix exponential. We then integrate the stochastic state, project it into the latent coordinates, and distinguish a valid covariance from the nonsingular covariance required by a Gaussian density.

A.1 CONDITIONING AND ASSUMPTIONS

We condition throughout on the parameters returned for one component. For CMD-X, these parameters depend on the history and component code; for CMD-D, they also depend on the terminal draw and requested grid. Given this conditioning, we work on a fixed finite interval $[0, H]$. The cumulative fields R_k and Ω_k are absolutely continuous, with $R_k(0) = \Omega_k(0) = 0$. We allow an absolutely continuous, nondecreasing energy G_k with $G_k(0) = p_k^0 \geq 0$, and define the process-noise rate

$$q_k(t) = e^{-2R_k(t)} G_k'(t) \geq 0 \quad \text{almost everywhere.} \tag{15}$$

On the finite interval this rate is integrable: R_k is bounded and G_k' is integrable. The initial Gaussian noises and Brownian motions are mutually independent across modes and latent coordinates, and independent of the conditioned parameters. The prediction heads employ a constant $q_k > 0$ directly recovering (7). Allowing $q_k(t)$ proves the kernel result for any admissible energy, including the fixed-anchor numerical construction in Appendix B.

A.2 WHY THE TRANSITIONS COMMUTE

Lemma 1 (Commuting rotation–scaling transitions)**.** *Let $A(t) = -\rho(t)I_2 + \omega(t)J$, where $J^2 = -I_2$ and ρ, ω are integrable. Then $A(t)A(s) = A(s)A(t)$ almost everywhere, and the fundamental transition is*

$$\Phi(t, s) = \exp\left(\int_s^t A(v)\, dv\right) = e^{-[R(t)-R(s)]}\mathrm{Rot}(\Omega(t) - \Omega(s))\,. \tag{16}$$

It satisfies $\Phi(t, u)\Phi(u, s) = \Phi(t, s)$ for $s \leq u \leq t$.

Proof. We suppress the mode index. Multiplying two members of the span of I_2 and J gives

$$(aI_2 + bJ)(a'I_2 + b'J) = (aa' - bb')I_2 + (ab' + a'b)J.$$

The right-hand side is unchanged when the factors are exchanged. Hence every pair of drift matrices commutes. Their integral is

$$M(t, s) = \int_s^t A(v)\, dv = -[R(t) - R(s)]I_2 + [\Omega(t) - \Omega(s)]J.$$

The two terms commute, so their exponentials factor. Expanding the exponential of J and using $J^2 = -I_2$ yields

$$e^{\theta J} = \left(\sum_{j=0}^{\infty} \frac{(-1)^j \theta^{2j}}{(2j)!}\right) I_2 + \left(\sum_{j=0}^{\infty} \frac{(-1)^j \theta^{2j+1}}{(2j+1)!}\right) J = \cos\theta\, I_2 + \sin\theta\, J = \mathrm{Rot}(\theta).$$

Consequently, the proposed transition has the form in (16). Differentiating it almost everywhere gives

$$\partial_t \Phi(t, s) = [-\rho(t)I_2 + \omega(t)J]\Phi(t, s) = A(t)\Phi(t, s), \qquad \Phi(s, s) = I_2.$$

Uniqueness of the linear initial-value problem establishes it as the fundamental transition (Rugh, 1996). Ultimately, decay increments and phase increments accumulate across adjacent intervals, proving the composition identity. □

Scope of the construction. The lemma remains valid for time-varying rates because their matrices remain in the same commuting algebra. A general time-varying linear system generally lacks this property. Among commuting real 2×2 families containing a matrix with nonreal eigenvalues, the rotation–scaling form is unique up to a fixed real similarity: the centralizer of a nonscalar 2×2 matrix is its span with I_2. A nonorthogonal similarity also transforms the noise covariance. Thus isotropic planar noise is an additional choice in our stochastic construction, rather than a consequence of commutativity.

A.3 From the stochastic state to the temporal kernel

Proof of Proposition 1. **State solution and mean.** For one mode and coordinate, variation of constants gives

$$\xi_{kd}(t) = \Phi_k(t,0)\xi_{kd}(0) + \int_0^t \Phi_k(t,v)\sqrt{q_k(v)}\, dW_{kd}(v). \tag{17}$$

The stochastic integral has zero mean, and $\mathbb{E}\xi_{kd}(0) = e_{\text{init}}$. Hence

$$\mathbb{E}\xi_{kd}(t) = e^{-R_k(t)}\mathrm{Rot}(\Omega_k(t))e_{\text{init}} = e^{-R_k(t)}\begin{pmatrix}\cos\Omega_k(t)\\ \sin\Omega_k(t)\end{pmatrix}.$$

Multiplying by $\alpha\ell_{kd}^\top$ and summing over k yields (6). The centered states at any finite set of times are linear functions of jointly Gaussian initial noise and Brownian integrals, so they form a joint Gaussian vector.

Marginal variance. The initial noise is independent of subsequent Brownian increments. Applying the Itô isometry to (17) therefore gives

$$\begin{aligned}
\mathrm{Var}\,\xi_{kd}(t) &= p_k^0\Phi_k(t,0)\Phi_k(t,0)^\top + \int_0^t q_k(v)\Phi_k(t,v)\Phi_k(t,v)^\top dv \\
&= \left[p_k^0 e^{-2R_k(t)} + \int_0^t q_k(v)e^{-2[R_k(t)-R_k(v)]}\, dv\right] I_2 \\
&= e^{-2R_k(t)}\left[p_k^0 + \int_0^t e^{2R_k(v)}q_k(v)\, dv\right] I_2 \\
&= e^{-2R_k(t)}G_k(t)I_2.
\end{aligned}$$

The second line uses $\mathrm{Rot}(\theta)\mathrm{Rot}(\theta)^\top = I_2$; the final line uses (15) and $G_k(0) = p_k^0$. Therefore rotations change temporal alignment but do not change a mode's isotropic marginal variance.

Cross-time covariance. For $t \geq s$, split the integral at s and use the transition composition identity:

$$\xi_{kd}(t) = \Phi_k(t,s)\xi_{kd}(s) + \nu_{kd}(t,s), \qquad \nu_{kd}(t,s) = \int_s^t \Phi_k(t,v)\sqrt{q_k(v)}\, dW_{kd}(v).$$

The innovation $\nu_{kd}(t,s)$ is independent of $\xi_{kd}(s)$. Its variance is

$$\begin{aligned}
\mathrm{Var}\,\nu_{kd}(t,s) &= \left[\int_s^t e^{-2[R_k(t)-R_k(v)]}q_k(v)\, dv\right] I_2 \\
&= e^{-2R_k(t)}[G_k(t) - G_k(s)]I_2 = V_k(t,s)I_2,
\end{aligned}$$

which proves the innovation formula in (10). The cross-time covariance follows as

$$\begin{aligned}
\mathrm{Cov}(\xi_{kd}(t), \xi_{kd}(s)) &= \Phi_k(t,s)\,\mathrm{Var}\,\xi_{kd}(s) \\
&= e^{-R_k(t)-R_k(s)}G_k(s)\mathrm{Rot}(\Omega_k(t) - \Omega_k(s)).
\end{aligned}$$

For $t < s$, transpose the expression with the arguments exchanged. Both cases therefore use $G_k(\min\{t,s\})$.

Projection and positive semidefiniteness. For a real loading $\ell = (c,b)^\top$, skew-symmetry of J implies $\ell^\top J\ell = 0$. Therefore

$$\ell^\top\mathrm{Rot}(\theta)\ell = \ell^\top(\cos\theta\, I_2 + \sin\theta\, J)\ell = (c^2+b^2)\cos\theta.$$

Independent modes contribute additively to the covariance, and independent state copies give zero covariance between distinct latent coordinates. Applying the last identity yields exactly (8). To establish positive semidefiniteness, take arbitrary finitely many query-coordinate pairs (t_r, d_r) and real coefficients a_r. Then

$$\sum_{r,s} a_r a_s \mathcal{K}_{d_r d_s}(t_r,t_s) = \mathrm{Var}\left(\sum_r a_r z_{d_r}(t_r)\right) \geq 0.$$

Consequently every finite kernel matrix is positive semidefinite, including cases where individual cosine factors are negative. The covariance is valid because the full matrix derives from a Gaussian process, not because every entry is nonnegative. □

A one-mode correlation identity. For an active scalar mode, divide its covariance by the square root of its two marginal variances. When $s \le t$ and both variances are positive, the loading, scale, and damping factors cancel:

$$\frac{\alpha^2\|\ell\|^2 e^{-R(t)-R(s)}G(s)\cos(\Omega(t)-\Omega(s))}{\alpha^2\|\ell\|^2 e^{-R(t)-R(s)}\sqrt{G(t)G(s)}} = \sqrt{\frac{G(s)}{G(t)}}\cos(\Omega(t)-\Omega(s)).$$

This proves (9). A sum of modes typically exhibits a weighted combination of these effects, rather than this single-mode factorization.

A.4 Positive definiteness and numerical conditioning

Positive semidefiniteness ensures a Gaussian law, which may be supported on a lower-dimensional subspace. The standard full-grid log density in (11) further requires positive-definite covariance on the supervised indices. We derive sufficient conditions for the independent-coordinate model employed in the experiments.

Fix distinct sorted queries $0 \le t_1 < \cdots < t_h$. Suppose $\alpha \neq 0$ and, for each coordinate d, there is a mode k with $\ell_{kd} \neq 0$, $p_k^0 > 0$, and $V_k(t,s) > 0$ whenever $t > s$. Consider a nonzero linear combination $\sum_r a_r z_d(t_r)$, and let r_* be its last nonzero coefficient. If $r_* > 1$, the selected mode's Gaussian innovation on $(t_{r_*-1}, t_{r_*}]$ occurs only in the final retained term. It is independent of the preceding states and has contribution to the variance

$$a_{r_*}^2\alpha^2\|\ell_{kd}\|^2 V_k(t_{r_*}, t_{r_*-1}) > 0.$$

If $r_* = 1$, its variance is positive because the selected mode exhibits positive initial variance and a nonzero loading. Every nonzero linear combination therefore yields positive variance. Each coordinate block is positive definite, and their block-diagonal concatenation is positive definite as well. Constant $q_k > 0$ injects strictly positive innovation variances on every nonzero interval; the numerical recurrence in Appendix B does so at distinct queries.

No bound relating d_z to $2K$ is required here, because each coordinate has independent state copies. Repeated occurrences of the same time-coordinate pair represent the same random variable and must be identified before evaluating a nonsingular density. Zero output scale or inactive loadings can instead induce a degenerate law, even though the kernel remains positive semidefinite.

Numerical jitter and likelihood scope. Positive definiteness in exact arithmetic does not guarantee good numerical conditioning: small output scales, small loadings, or closely spaced queries can render Cholesky factorization sensitive to roundoff. Introducing δI with $\delta > 0$ stabilizes the finite-grid covariance, but evaluates the density of the perturbed law $\mathcal{N}(m, K + \delta I)$. If that law is also sampled, the samples must include matching independent noise. Jitter does not imply positive definiteness of the original K, and a positive numerical nugget is distinct from the PSD guarantee in Proposition 1. Appendix H.2 specifies the relative jitter and matched sampling employed in the synthetic likelihood experiment.

A.5 Shared-state extension

The experimental model utilizes independent state copies across latent coordinates. A related extension shares each modal state and thereby introduces cross-coordinate covariance within a component. Let

$$z(t) = \alpha\sum_k C_k\xi_k(t), \qquad C_k = [c_k\ b_k] \in \mathbb{R}^{d_z\times 2},$$

where c_k and b_k are loading vectors and the states remain independent across modes. Applying the state covariance derived above yields

$$\mathcal{K}^{\text{shared}}(t,s) = \alpha^2\sum_k e^{-R_k(t)-R_k(s)}G_k(\min(t,s))C_k\mathrm{Rot}(\Omega_k(t)-\Omega_k(s))C_k^\top. \tag{18}$$

To see its cross-coordinate structure, write $\Delta\Omega_k = \Omega_k(t) - \Omega_k(s)$ and expand

$$\begin{aligned} C_k\mathrm{Rot}(\Delta\Omega_k)C_k^\top &= \cos(\Delta\Omega_k)(c_kc_k^\top + b_kb_k^\top) + \sin(\Delta\Omega_k)C_kJC_k^\top \\ &= \cos(\Delta\Omega_k)(c_kc_k^\top + b_kb_k^\top) + \sin(\Delta\Omega_k)(b_kc_k^\top - c_kb_k^\top). \end{aligned}$$

The sine term is skew-symmetric in its coordinate indices and changes sign when the times are interchanged, guaranteeing $\mathcal{K}^{\text{shared}}(t,s) = \mathcal{K}^{\text{shared}}(s,t)^\top$. Positive semidefiniteness again follows by computing the covariance of the projected Gaussian states.

At a single time, stack the loadings as $C = [C_1 \ \cdots \ C_K]$. Then

$$\mathcal{K}^{\text{shared}}(t,t) = \alpha^2 C \ \text{blockdiag}_k(\Lambda_k(t) I_2)\, C^\top,$$

hence its rank is at most $\min(d_z, 2K)$. For a nonsingular per-time covariance, the stacked loading must have full row rank, which demands $d_z \le 2K$. Full row rank, nonzero α, positive initial modal variances, and positive modal innovations on every nonzero interval also suffice for a nonsingular joint covariance at distinct queries: apply the last-nonzero-time argument above to vector coefficients and the full-rank projected innovation covariance. This rank requirement belongs to the shared-state extension; it does not extend to our independent-coordinate construction.

A.6 Mean envelope and fixed-pole special case

For $\rho_k(t) \ge \rho_{\min} > 0$, integration gives $R_k(t) \ge \rho_{\min} t$. By Cauchy–Schwarz,

$$|c_{kd}\cos\Omega_k(t) + b_{kd}\sin\Omega_k(t)| \le \sqrt{c_{kd}^2 + b_{kd}^2}\sqrt{\cos^2\Omega_k(t) + \sin^2\Omega_k(t)} = \sqrt{c_{kd}^2 + b_{kd}^2}.$$

The triangle inequality in (6) therefore yields

$$|m_d(t)| \le |\alpha| e^{-\rho_{\min} t} \sum_k \sqrt{c_{kd}^2 + b_{kd}^2}. \tag{19}$$

This is a bound for the latent component mean conditional on its returned parameters. Stochastic variance is governed separately by G_k, and the observation decoder is nonlinear, hence the bound does not imply exponential decay of observation-space forecast uncertainty.

Setting the chirp coefficients to zero gives $R_k(t) = \rho_k^{\text{f}} t$ and $\Omega_k(t) = \omega_k^{\text{f}} t$. The mean then becomes the fixed-pole modal sum. For constant process-noise rate and $\rho_k^{\text{f}} > 0$, the energy integral can also be evaluated explicitly:

$$G_k(t) = p_k^0 + q_k \int_0^t e^{2\rho_k^{\text{f}} v}\, dv = p_k^0 + \frac{q_k}{2\rho_k^{\text{f}}}(e^{2\rho_k^{\text{f}} t} - 1),$$
$$\Lambda_k(t) = e^{-2\rho_k^{\text{f}} t} G_k(t) = p_k^0 e^{-2\rho_k^{\text{f}} t} + \frac{q_k}{2\rho_k^{\text{f}}}(1 - e^{-2\rho_k^{\text{f}} t}).$$

Hence $\Lambda_k(t) \to q_k/(2\rho_k^{\text{f}})$ as $t \to \infty$. If the initial variance equals this limit, the corresponding scalar modal kernel is stationary:

$$\mathcal{K}_{k,d}(t,s) = \alpha^2 \|\ell_{kd}\|^2 \frac{q_k}{2\rho_k^{\text{f}}} e^{-\rho_k^{\text{f}}|t-s|} \cos(\omega_k^{\text{f}}(t-s)).$$

The fixed-pole controls thus maintain a stochastic temporal component while constraining frequency and decay within a forecast window. Chirping generalizes this family by enabling the accumulated phase and damping to evolve nonlinearly with time.

B Pole constraints and numerical covariance

Bounded pole fields. We use four signed cosine features at half-cycle and full-cycle harmonics of a time scale L. Writing their antiderivatives as $\bar{\phi}_m$, we parameterize

$$R_k(t) = \rho_k^{\text{f}} t + \sum_{m=1}^{4} a_{\rho,km}^2 \bar{\phi}_m(t), \qquad \Omega_k(t) = \omega_k^{\text{f}} t + \sum_{m=1}^{4} a_{\omega,km}^2 \bar{\phi}_m(t). \tag{20}$$

The squared coefficients denote the rescaled values defined below. With $\rho_k^{\text{f}} \in (\rho_{\min}, 4/L)$ and $\omega_k^{\text{f}} \in (0, \pi)$, we obtain

$$\rho_{\min} \le \rho_k(t) \le 2\rho_k^{\text{f}} - \rho_{\min}, \qquad 2\omega_k^{\text{f}} - \pi \le \omega_k(t) \le \pi. \tag{21}$$

Thus damping remains positive and $|\omega_k(t)| \le \pi$ in model-time units. The instantaneous frequency can have either sign. The upper bound on the full decay field is $2\rho_k^{\mathrm{f}} - \rho_{\min} < 8/L - \rho_{\min}$, while $4/L$ bounds its constant term. Zero chirp coefficients recover the fixed-pole family.

Signed harmonic basis and coefficient rescaling. We set $f_m = (1/2, 1/2, 1, 1)$ and $s_m = (+1, -1, +1, -1)$, with

$$\phi_m(t) = s_m \cos(2\pi f_m t/L), \qquad \bar{\phi}_m(t) = \frac{s_m \sin(2\pi f_m t/L)}{2\pi f_m/L}.$$

For raw coefficient a_m and available headroom h, we use

$$\widetilde{a}_m^2 = \frac{a_m^2 h}{\sum_{m'} a_{m'}^2 + h}, \qquad h_\rho = \rho^{\mathrm{f}} - \rho_{\min}, \quad h_\omega = \pi - \omega^{\mathrm{f}}.$$

These nonnegative coefficients sum to at most h. Since $|\phi_m(t)| \le 1$, the perturbation lies in $[-h, h]$, proving (21). The signed basis supports positive and negative deviations from each constant pole despite the squared coefficients. The frequency constraint is a bound in model-time units; the query offsets determine the physical sampling gaps.

Equivalent centered basis. The centered notation used in the drift ablation writes each field as

$$\rho_k(t) = \rho_{\mathrm{floor},k} + \sum_m \widetilde{a}_{\rho,km}^2(\phi_m(t) - 1), \qquad \omega_k(t) = \omega_{\mathrm{floor},k} + \sum_m \widetilde{a}_{\omega,km}^2(\phi_m(t) - 1).$$

Expanding the brackets recovers the uncentered fields in (20) with unchanged coefficients and adjusted constant terms

$$\rho_k^{\mathrm{f}} = \rho_{\mathrm{floor},k} - \sum_m \widetilde{a}_{\rho,km}^2, \qquad \omega_k^{\mathrm{f}} = \omega_{\mathrm{floor},k} - \sum_m \widetilde{a}_{\omega,km}^2.$$

The integrated fields align as well, since integrating $\phi_m(t) - 1$ yields $\bar{\phi}_m(t) - t$. These are two representations for the same pole field after the intercept adjustment. The bounds in (21) apply to the adjusted constant terms ρ_k^{f} and ω_k^{f}; the centered intercepts denoted by "floor" are not interchangeable with those terms in the inequalities.

For each latent coordinate, reversing the sign of the complete phase function leaves the kernel invariant. Reversing the sine loading at the same time preserves the mean. This symmetry concerns a global phase reversal; a varying frequency field can induce time-dependent phase differences that a constant loading cannot absorb.

Positive variance recurrence. We sort the query offsets and write $\Delta R_r = R(t_r) - R(t_{r-1})$ and $\Delta t_r = t_r - t_{r-1}$. Starting at the last-observation origin $t_0 = 0$, we propagate

$$\widehat{\Lambda}_r = e^{-2\Delta R_r}\widehat{\Lambda}_{r-1} + \widehat{V}_r, \qquad \widehat{\Lambda}_0 = p^0. \tag{22}$$

Define

$$\psi(x) = \frac{1 - e^{-2x}}{2x}, \qquad \psi(0) = 1, \qquad \widehat{V}_r = q\Delta t_r \psi(\Delta R_r).$$

We evaluate ψ with a stable small-x branch. The numerical cell innovation $\widehat{V}_r \ge 0$ approximates the exact $V(t_r, t_{r-1})$ in (10). Combining (22) with the analytic transition specifies the finite Gaussian law utilized by CMD-D, CMD-X, and the fixed-pole mixtures. The recurrence computes the variance integral exactly when R is linear on each integration interval and approximates it for smoothly drifting R. We sample exactly from the resulting numerical Gaussian law.

Why the recurrence preserves a valid covariance. For an interval of length Δt, replacing $R(v) - R(t_{r-1})$ by its linear interpolant $\Delta R_r(v - t_{r-1})/\Delta t$ gives

$$\widehat{V}_r = q\Delta t \int_0^1 e^{-2\Delta R_r(1-u)}\, du = q\Delta t\, \psi(\Delta R_r)$$

under that interpolant. This establishes exactness for linear R on the interval and explains the approximation for a drifting decay field. Moreover, defining $\widehat{G}_r = e^{2R(t_r)}\widehat{\Lambda}_r$ in (22) yields

$$\widehat{G}_r - \widehat{G}_{r-1} = q\Delta t_r e^{2R(t_r)}\psi(\Delta R_r) \ge 0.$$

Therefore the numerical energies are nondecreasing on the chosen grid. The resulting initial Gaussian state and independent Gaussian innovations induce a positive-semidefinite covariance directly. For $q > 0$ and distinct query times, every numerical innovation variance is strictly positive. The transition is analytic, the variance integral is approximated, and joint sampling is exact for the derived finite-grid law.

Joint path sampling. We choose one mixture component for the complete path, draw independent modal initial states for each latent coordinate, and propagate

$$\xi_{kd}(t_r) = e^{-\Delta R_{kr}} \mathrm{Rot}(\Delta\Omega_{kr})\xi_{kd}(t_{r-1}) + \sqrt{\widehat{V}_{kr}}\,\eta_{kdr}, \quad \eta_{kdr} \sim \mathcal{N}(0, I_2).$$

We apply the loading and output scale to extract the latent values and restore the requested query order. This propagation maintains both the shared component identity and the component's residual temporal covariance.

B.1 A FIXED-ANCHOR CONSTRUCTION

We can evaluate a query-independent energy by fixing the domain $[0, H]$ and anchors $0 = a_0 < \cdots < a_A = H$ before receiving queries. Here A is any fixed positive number of integration intervals; the construction is independent of a particular predictor configuration. We apply (22) at the anchors and, for $t \in [a_n, a_{n+1}]$, evaluate

$$\widehat{\Lambda}(t) = e^{-2[R(t)-R(a_n)]}\widehat{\Lambda}(a_n) + q(t - a_n)\psi(R(t) - R(a_n)). \tag{23}$$

Here H is specified independently of the requested query set, and L governs the harmonic time scale. Maintaining the origin, domain, and anchors fixed enables us to evaluate one covariance function on different subsets of times.

Proposition 2 (Monotone off-grid energy)**.** *Suppose $q \geq 0$, $p^0 \geq 0$, R is absolutely continuous and nondecreasing, and the anchor grid is fixed. The anchor recursion and* (23) *define a single continuous, nondecreasing energy function $\widehat{G}(t) = e^{2R(t)}\widehat{\Lambda}(t)$.*

Proof. Within a cell, let $x = R(t) - R(a_n) \geq 0$. Then

$$\widehat{G}(t) - \widehat{G}(a_n) = qe^{2R(a_n)}(t - a_n)\,\mathrm{exprel}(2x), \qquad \mathrm{exprel}(y) = \int_0^1 e^{uy}\,du.$$

Both $t - a_n$ and $\mathrm{exprel}(2x)$ are nonnegative and nondecreasing in t, hence the increment is nondecreasing throughout the cell. At its right endpoint the expression aligns with the next anchor's recursion. This proves continuity and monotonicity across cells. The $x = 0$ limit is continuous, and absolute continuity on the finite partition follows from that of R. □

Corollary 1 (Projective component laws)**.** *Suppose component parameters depend only on history, and the time origin, domain, anchors, and off-grid energy are specified independently of queries. Substituting $\widehat{G}$ into* (8) *induces a PSD kernel whose finite Gaussian restrictions are consistent under subsets and permutations. A fixed finite mixture with history-only weights is also consistent. A decoder operating separately at each time retains consistency under time restriction, and subsequent observed-channel selection preserves the corresponding marginals.*

Proof. Propositions 1 and 2 yield a valid covariance function. Restricting the mean and covariance to a subset extracts its Gaussian marginal. Finite mixing commutes with restriction when the weights and components are invariant. A per-time decoder commutes with deleting time points, and channel selection is a subsequent measurable projection. □

The positive decay bound makes R nondecreasing, as required by Proposition 2. For arbitrary $s \leq t$, the innovation variances of this fixed-anchor law are

$$\widehat{V}_k(t, s) = e^{-2R_k(t)}[\widehat{G}_k(t) - \widehat{G}_k(s)] \geq 0.$$

Using these increments in the modal sampler generates joint draws at arbitrary sorted queries. For pairs spanning anchor cells, this energy difference must be used rather than a new endpoint approximation $q_k(t - s)\psi(R_k(t) - R_k(s))$, which need not agree with the fixed-anchor law. Corollary 1 specifies the additional query-independence conditions for cross-grid consistency. The query-grid recurrence utilized by the mixture predictors defines their forecast law on each requested grid; the fixed-anchor construction evaluates one energy function across grids.

C MODEL SPECIFICATIONS AND TRAINING

Training objectives. We trained the joint denoising conditional in CMD-D with full-grid GP NLL and trained CMD-X and the fixed-pole controls with full-grid mixture NLL. Table 4 outlines the training and prediction specifications. Each component likelihood incorporates temporal covariance within every latent coordinate. We train the denoiser at one sampled noise level per example and derive its forecast mixture from reverse trajectories at inference.

Table 4: **Training and prediction specification.** Component likelihoods couple query times within each latent coordinate; a common component identity couples coordinates in the mixture. The frozen representation is shared by the CMD-D, CMD-X, and fixed-pole predictors shown here.

Item	CMD-D	CMD-X and fixed-pole mixtures
History	Full frozen summary at each step	Same frozen summary
Other conditioning	Noisy latent z_τ, diffusion level τ	Learned component code
Parameter generator	64-step deterministic DDIM map of Gaussian terminal noise	Shared MLP with learned codes
Stage-2 objective	Denoising NLL, (13)	Mixture NLL, (12)
Weights	Uniform	Uniform
Component index	Exchangeable terminal-draw order	Fixed learned-code order
Query-grid evaluation	Latent grid; mean and covariance	Mean synthesis and covariance
Variance integration	Query-grid recurrence	Same recurrence
Forecast samples	Joint modal Markov draws	Joint modal Markov draws
Stage-2 learning rate	10^{-4}	3×10^{-4}
Final-stage limit	200 epochs, best validation objective	200 epochs, best validation objective

Optimization and initialization. Final-stage limits and checkpoint selection are specified in Appendix K. We optimized the direct heads with AdamW with learning rate 3×10^{-4}, weight decay 10^{-4}, cosine annealing, and gradient-norm clipping at 1.0. We initialized component codes from $\mathcal{N}(0, 0.02^2)$ and the parameter readout from $\mathcal{N}(0, 0.01^2)$ with zero bias. Raw chirp coefficients have initial standard deviation 10^{-4}, enabling gradients through their squared parameterization. The scalar α multiplies the mean once and the covariance quadratically, and we learn it with constraint $|\alpha| \leq 1$. For CMD-D, we warm-started the MSE-trained denoiser and trained with MinSNR weighting at $\gamma = 4.5$ (Hang et al., 2023). We initialized its variance from the latent residual scale and trained the variance head jointly with the denoiser. The VAE and history summarizer remained frozen.

Table 5: **Dataset-dependent configuration.** L is the harmonic time scale in model-time units.

Quantity	NOAA-UK	USHCN	BMS-air	PhysioNet
Queries h	168	12	168	12
Latent width d_z	16	16	24	16
Harmonic scale L	168	12	168	12
CMD-D modes K	64	64	64	64
CMD-X / equal-K LTI modes	64	64	64	64
Larger LTI modes K_*	79	79	74	79
Batch size	64	256	128	64

Matching trainable capacity. We compare heads on a common frozen representation and count their trainable parameters separately. With $M_\phi = 4$ harmonic coefficients per pole field, the approximate matching rule is $K_* \approx K(1 + M_\phi/(d_z + 1))$. For $(K, d_z) = (64, 16)$ this yields 79.05, and we use 79 fixed-pole modes.

Frozen representations. The shared frozen pipeline applies to CMD-X, CMD-D, LLapDiff, the fixed-pole heads, and Chirp-MSE. The additional diagonal-Gaussian, low-rank Gaussian, and Student-t heads are distributional controls structured around the same encoder architecture. QuITE++, TFMixer, SDE-VI, MOSES, ProFITi, CRU, NCDSSM, and PatchTST instead train their own models on raw observations and do not leverage CMD's frozen VAE or summarizer.

Probabilistic baseline variants. The benchmarked QuITE++ and TFMixer variants employ diagonal-Gaussian (DG) output heads, each forecast coordinate admits a predicted mean and posi-

tive standard deviation. A DG head factorizes the conditional residual distribution across forecast coordinates, while the backbone aggregates information across the observed history.

Decay and time conventions. We set the lower decay bound to 10^{-3} for the mixture heads. Equation (21) gives the bounds on the full pole fields. The harmonic scale L and query count h have distinct roles: L governs the feature frequencies in model-time units, while the native query offsets specify the forecast times and preserve the lead gap from the last observation. The integration domain H in the fixed-anchor construction is a separate quantity.

C.1 Training and forecasting algorithms

Algorithms 1 and 2 outline the stage-2 procedures we implemented after freezing the representation. Both algorithms evaluate component densities on the supervised query indices and retain a single component identity within each sampled path. The likelihood exploits the complete temporal covariance of each latent coordinate. Sorting queries is a numerical operation; we restore their requested order before decoding.

Common synthesis and sampling. Given a generated parameter set Θ, we evaluate the constrained pole fields, compute the mean by (6), and propagate the modal variances by (22). These variances and the analytic transitions determine the finite-grid covariance. To draw a path from that component, we initialize independent states $\xi_{kd}(0) \sim \mathcal{N}(e_{\text{init}}, p_k^0 I_2)$, apply the joint propagation in Appendix B, and project the states with $\alpha \ell_{kd}^\top$. This operation samples residual temporal dependence without a dense covariance factorization. We denote it by $\mathrm{JointDraw}(\Theta, Q)$.

The model law has uniform mixture weights. The algorithms deliver a finite scoring ensemble by the deterministic allocation in Appendix D.

Algorithm 1 CMD-X: direct mixture training and forecasting

Require: Frozen history summarizer and VAE; training/validation windows; $M_{\text{train}} = 25$; requested Q; retained count $M_{\text{pool}} \le 25$; $S = 25$.
Initialize the shared head G_θ and learned codes $e_1, \dots, e_{25}$.
for each stage-2 training minibatch **do**
  Encode history as E and clean future as z; obtain supervised indices $\mathcal{O}_t$.
  Generate $\Theta^{(j)} = G_\theta(E, e_j)$ for every $j = 1, \dots, 25$ in one pass.
  Synthesize each component on Q; compute $\ell_j = \ell_Q(\Theta^{(j)}; z)$ by (11).
  Compute the batch mean of $-[\mathrm{logsumexp}_j\, \ell_j - \log 25]/n_{\mathcal{O}}$.
  Update θ and the component codes using this mixture loss, (12).
end for
Retain the checkpoint with the lowest validation mixture NLL during 200 epochs.
Forecasting with the retained checkpoint
Encode the supplied history and generate all 25 parameter sets in one pass.
Retain the first M_{pool} components; set $B = \lceil S/M_{\text{pool}} \rceil$.
for $j = 1, \dots, M_{\text{pool}}$ **do**
  Draw B independent paths using $\mathrm{JointDraw}(\Theta^{(j)}, Q)$.
end for
Concatenate paths in component order, retain the first S, and decode each time point.
return The S decoded paths and retained Gaussian component parameters.

Reverse-update convention. For CMD-D, let $\tau_{64} > \cdots > \tau_0 = 0$ denote the configured 64-step inference schedule and let $a_l = \bar{\alpha}_{\tau_l}$, with $a_0 = 1$. If the denoiser predicts clean mean $\widehat{z}_0$ from current state v_l, the deterministic DDIM update is

$$\widehat{\epsilon}_l = \frac{v_l - \sqrt{a_l}\,\widehat{z}_0}{\sqrt{1 - a_l}}, \qquad v_{l-1} = \sqrt{a_{l-1}}\,\widehat{z}_0 + \sqrt{1 - a_{l-1}}\,\widehat{\epsilon}_l. \tag{24}$$

Here the levels with a denoiser evaluation satisfy $0 < a_l < 1$; the clean endpoint requires no additional evaluation. The inferred component is the parameter set from the final denoiser evaluation. Its covariance supplies the subsequent Gaussian path draws; the deterministic reverse updates leverage its mean. The notation specifies the update without enforcing an additional spacing rule on the con-

figured noise schedule. Training uses the configured diffusion-level sampling and MinSNR weights w_γ, with $\gamma = 4.5$.

Algorithm 2 CMD-D: full-grid denoising training and component generation

Require: Frozen history summarizer and VAE; MSE-trained denoiser; training/validation windows; configured noise schedule and w_γ; requested Q; M_{pool}; $S = 25$.

Warm-start the denoiser and initialize the variance head from the latent residual scale.
for each stage-2 training minibatch **do**
  Encode history as E and clean future as z; obtain $\mathcal{O}_t$.
  Sample one diffusion level τ and independent $\epsilon \sim \mathcal{N}(0, I)$ per example.
  Form $z_\tau = \sqrt{\bar{\alpha}_\tau} z + \sqrt{1 - \bar{\alpha}_\tau}\epsilon$.
  Predict $\Theta = \Theta_\theta(z_\tau, \tau, E)$ and synthesize its component on Q.
  Compute the batch mean of $-w_\gamma(\tau)\ell_Q(\Theta; z)/n_{\mathcal{O}}$.
  Update the denoiser and variance head jointly using (13).
end for
Retain the checkpoint with the lowest validation denoising objective during 200 epochs.
Forecasting with the retained checkpoint
for $j = 1, \ldots, M_{\text{pool}}$ **do**
  Draw independent terminal noise $v_{64} = u_j \sim \mathcal{N}(0, I_{hd_z})$.
  for $l = 64, \ldots, 1$ **do**
    Predict $\Theta_l = \Theta_\theta(v_l, \tau_l, E)$; evaluate $\widehat{z}_0 = m_{\Theta_l, Q}$.
    Update v_{l-1} by the deterministic DDIM rule (24).
  end for
  Retain $\Theta^{(j)} = \Theta_1$ as the component returned by trajectory j.
end for
Set $B = \lceil S/M_{\text{pool}} \rceil$; draw B joint paths from each retained component.
Concatenate paths in component order, retain the first S, and decode each time point.
return The S decoded paths and the uniformly weighted component set.

D Data preparation and evaluation

Datasets and preprocessing. We evaluate NOAA-UK, USHCN, BMS-air, and PhysioNet. The three environmental datasets comprise 194,509, 350,665, and 414,710 windows, respectively, with 5/5/12 input channels, 4/5/11 target channels, and 7/1,114/12 entities. We preserve observation-time gaps and channel-wise missingness; USHCN additionally uses irregular forecast-query timestamps. Their mean per-timestamp observation coverage is 99.6/21.1/100%, with minima of 14.3/15.4/83.3%. PhysioNet comprises 12,497 windows from 1,022 entities, with four input channels and one target channel. We follow the PhysioNet preprocessing protocol of LLapDiff (You et al., 2026), employing a 24-step context and a 12-step forecast horizon. Observations are assigned to the nearest bin on the native-interval indexing grid without interpolation, preserving observation masks and time metadata; each window contains at least one observed forecast target. We apply chronological 70/10/20 training/validation/test splits before window construction and fit channel-wise standardization on the training split. For PhysioNet, remaining missing inputs are zero-filled after normalization and accompanied by observation masks.

Latent supervision. We provided the VAE encoder with $[y \odot o, o]$, setting missing values to zero and supplying their observation indicators. The latent supervision mask is

$$o_r^{\text{lat}} = \bigvee_{n,c} o_{rnc}, \qquad \mathcal{O}_t = \{r : o_r^{\text{lat}} = 1\}.$$

Here n and c index entities and channels. At each supervised time, all latent coordinates contribute to the full-grid GP NLL. We restrict the predicted mean and covariance to $\mathcal{O}_t$ and normalize the loss by $d_z|\mathcal{O}_t|$. This objective fits the distribution of encoded targets; we decode sampled latent paths to evaluate forecasting accuracy in observation space.

Forecast scores. We evaluated every method using $S = 25$ observation-space forecast paths. For an observed target vector y and samples $x_1, \ldots, x_S$, we computed

$$\mathrm{ES} = \frac{1}{S}\sum_{a=1}^{S} \|x_a - y\|_2 - \frac{1}{2S^2}\sum_{a,b=1}^{S} \|x_a - x_b\|_2.$$

We computed CRPS from the corresponding scalar expression and averaged over observed target entries. For one forecast window, let $t_1, \ldots, t_T$ be the ordered query times in the dataset's scoring time units, c index channels, and $o_{r,c}$ indicate an observed target. Define

$$\delta = \max\left\{10^{-12}, \mathrm{median}_{1\le r<T}\,|t_{r+1} - t_r|\right\}, \qquad w_{rs} = \begin{cases} 1/\max\{|t_r - t_s|, \delta\}, & r \neq s, \\ 0, & r = s. \end{cases} \tag{25}$$

With $\rho_p(u) = \max\{|u|, 10^{-12}\}^p$, path index a, and order $p = 0.5$, the implemented finite-ensemble score is

$$\mathrm{VS} = \sum_{c=1}^{D}\sum_{r,s=1}^{T} w_{rs} o_{r,c} o_{s,c}\left[\rho_p(y_{r,c} - y_{s,c}) - \frac{1}{S}\sum_{a=1}^{S}\rho_p(x_{a,r,c} - x_{a,s,c})\right]^2. \tag{26}$$

Only same-channel pairs participate, with uniform channel weights after channel standardization. Both orderings of each distinct pair are incorporated, and both endpoints must be observed. The per-window median consecutive gap bounds the weight of near-coincident queries; self-pairs have zero weight. The score is a sum with no normalization by the number of pairs, channels, or total weight. VS is comparable within the same dataset, horizon, time units, and scoring protocol. The plain Monte Carlo mean inside the square induces the usual finite-ensemble bias, so varying S also changes the estimator. The 10^{-12} difference clamp above specifies the numerical implementation.

PIT-ECE measures the departure of randomized empirical ranks from uniformity. We extracted central 80% intervals from the sampled marginal distributions and measured coverage and interval width. These scores describe the finite ensemble delivered by each method under the common 25-path evaluation budget.

Component pooling and sample allocation. For each direct head, we retain the first M_{pool} codes of a model trained with 25 components. For CMD-D, we generate M_{pool} components from separate terminal draws. Each retained component contributes $\lceil S/M_{\mathrm{pool}} \rceil$ joint paths; we concatenate them in component order and retain the first S. At $S = 25$, this gives one path per component for $M_{\mathrm{pool}} = 25$. For $M_{\mathrm{pool}} = 8$, the allocation is $(4, 4, 4, 4, 4, 4, 1, 0)$; for $M_{\mathrm{pool}} = 16$, twelve components contribute two paths, the next contributes one, and the remaining three contribute zero. Thus the small-M comparisons evaluate retained-code prefixes with this sample-allocation rule. Within each drawn path, component identity is fixed across all times and coordinates.

Validation and test evaluation. We select hyperparameters and checkpoints on validation data, then score the selected models on the test split. The headline setting is $M = 25$; the retained-pool sweep explores $M \in \{1, 2, 4, 8, 16, 25\}$. Appendix K outlines search and final training.

Variation across runs. We executed ten independent runs with seeds 0–9. The tables report their mean scores, and $\pm$ denotes the standard deviation across these runs. This variation reflects training randomness and is distinct from the Monte Carlo variation of path samples. Appendix H separately specifies the ten-seed synthetic experiment and its independent test batches.

E COMPLETE FORECASTING RESULTS

Tables 6–9 present the full component-count sweeps complementing the main paper. M quantifies retained components for the direct heads and generated components for CMD-D; every score leverages $S = 25$ decoded paths. The direct sweeps retain prefixes of a single 25-code model. Values after $\pm$ are standard deviations across ten runs with seeds 0–9. Lower ES, VS, CRPS, and PIT-ECE are better. Central interval coverage targets 0.80; width is interpreted jointly with coverage. DG denotes a diagonal-Gaussian output adaptation. Tables 10–13 report additional marginal, joint, and calibration metrics for the forecasting baselines and same-checkpoint controls.

Calibration and interval diagnostics. The calibration diagnostics exhibit different tradeoffs across datasets. On NOAA-UK, CMD-D yields higher coverage than Chirp-MSE with wider intervals. On USHCN, CMD-X achieves lower PIT-ECE, while CMD-D's coverage is closer to the nominal 0.80. On PhysioNet, CMD-D produces lower joint scores, while CMD-X exhibits lower CRPS and PIT-ECE, narrower intervals, and coverage closer to 0.80. The tables retain the complete interval and calibration results for all methods.

Table 6: **NOAA-UK: complete component sweep.** Mean $\pm$ standard deviation across ten seeds (0–9); lower scores are better for error metrics (best results in bold, second-best results underlined). Coverage targets 0.80. Direct and LTI rows retain the first M codes of a 25-component head; CMD-D generates M components. LTI subscripts give the mode count K. Every score uses 25 paths.

M	Model	ES	VS	CRPS	PIT-ECE	Coverage	Width
1	CMD-D	18.785±0.576	<u>556.82</u>±28.10	0.6691±0.0130	0.2744±0.0709	0.404±0.037	1.113±0.081
1	CMD-X	**14.128**±0.328	**518.50**±18.58	**0.4658**±0.0178	<u>0.1145</u>±0.0348	0.585±0.039	1.239±0.055
1	LTI_{64}	<u>16.323</u>±0.135	888.94±67.39	<u>0.5677</u>±0.0042	**0.0979**±0.0043	0.553±0.026	1.541±0.109
1	LTI_{79}	16.400±0.129	905.98±87.29	0.5720±0.0035	0.1512±0.0193	0.547±0.021	1.560±0.104
2	CMD-D	<u>16.045</u>±0.318	**497.10**±21.57	0.5735±0.0079	0.2306±0.0622	0.559±0.031	1.508±0.086
2	CMD-X	**14.130**±0.267	<u>517.65</u>±19.99	**0.4659**±0.0163	<u>0.1184</u>±0.0332	0.586±0.034	1.247±0.063
2	LTI_{64}	16.442±0.105	898.35±60.34	<u>0.5683</u>±0.0032	**0.0989**±0.0059	0.557±0.024	1.558±0.099
2	LTI_{79}	16.430±0.100	898.32±59.39	0.5687±0.0029	0.1280±0.0175	0.552±0.017	1.557±0.075
4	CMD-D	<u>14.836</u>±0.234	**466.43**±18.38	<u>0.5274</u>±0.0066	0.2014±0.0392	0.656±0.017	1.732±0.092
4	CMD-X	**13.840**±0.213	<u>517.85</u>±21.97	**0.4596**±0.0075	<u>0.1156</u>±0.0285	0.615±0.034	1.259±0.068
4	LTI_{64}	16.351±0.103	901.05±52.56	0.5684±0.0033	**0.0976**±0.0042	0.559±0.026	1.565±0.107
4	LTI_{79}	16.308±0.112	894.61±54.59	0.5678±0.0039	0.1251±0.0134	0.557±0.013	1.561±0.056
8	CMD-D	<u>14.287</u>±0.211	**459.20**±16.19	<u>0.5073</u>±0.0061	0.1444±0.0343	0.689±0.015	1.820±0.093
8	CMD-X	**13.698**±0.171	<u>520.93</u>±21.49	**0.4574**±0.0066	**0.1036**±0.0268	0.661±0.028	1.277±0.062
8	LTI_{64}	16.286±0.103	892.36±61.77	0.5663±0.0032	<u>0.1218</u>±0.0068	0.561±0.026	1.556±0.113
8	LTI_{79}	16.255±0.120	884.53±73.44	0.5657±0.0034	0.1261±0.0121	0.562±0.015	1.562±0.080
16	CMD-D	<u>13.914</u>±0.178	**450.18**±10.92	<u>0.4928</u>±0.0045	<u>0.1157</u>±0.0327	0.718±0.011	1.868±0.098
16	CMD-X	**13.704**±0.147	<u>519.77</u>±21.36	**0.4581**±0.0051	**0.1092**±0.0221	0.629±0.025	1.280±0.062
16	LTI_{64}	16.318±0.108	906.02±61.55	0.5671±0.0031	0.1199±0.0064	0.566±0.025	1.580±0.114
16	LTI_{79}	16.259±0.138	889.31±69.39	0.5656±0.0042	0.1234±0.0099	0.563±0.013	1.569±0.070
25	CMD-D	**13.608**±0.159	**438.80**±8.16	**0.4503**±0.0035	**0.0943**±0.0308	0.731±0.011	1.940±0.079
25	CMD-X	<u>13.771</u>±0.168	<u>518.61</u>±20.98	<u>0.4593</u>±0.0079	0.1159±0.0213	0.624±0.027	1.278±0.061
25	LTI_{64}	16.345±0.116	915.35±58.85	0.5681±0.0035	<u>0.1077</u>±0.0053	0.569±0.025	1.594±0.110
25	LTI_{79}	16.268±0.128	894.77±79.24	0.5661±0.0038	0.1119±0.0079	0.564±0.017	1.573±0.090

Table 7: **USHCN: complete component sweep.** Mean ± standard deviation across ten seeds (0–9); lower scores are better for error metrics (best results in bold, second-best results underlined). Coverage targets 0.80. Direct and LTI rows retain the first M codes of a 25-component head; CMD-D generates M components. LTI subscripts give the mode count K. Every score uses 25 paths.

M	Model	ES	VS	CRPS	PIT-ECE	Coverage	Width
1	CMD-D	2.3383±0.2755	0.6960±0.0522	<u>0.3879</u>±0.0294	<u>0.1504</u>±0.0152	0.329±0.032	0.861±0.136
1	CMD-X	**1.5724**±0.0211	**0.4934**±0.0350	**0.2791**±0.0089	**0.0771**±0.0135	0.599±0.034	0.562±0.092
1	LTI_{64}	2.2637±0.0952	0.6010±0.0316	0.4109±0.0275	0.2161±0.0370	0.475±0.053	0.574±0.044
1	LTI_{79}	<u>2.1800</u>±0.2344	<u>0.5846</u>±0.0188	0.3920±0.0377	0.1970±0.0257	0.516±0.025	0.634±0.157
2	CMD-D	<u>1.9146</u>±0.1697	<u>0.5671</u>±0.0186	<u>0.3202</u>±0.0111	<u>0.1222</u>±0.0173	0.451±0.025	0.982±0.084
2	CMD-X	**1.5678**±0.0233	**0.4743**±0.0128	**0.2718**±0.0102	**0.0694**±0.0084	0.632±0.036	0.574±0.080
2	LTI_{64}	2.2623±0.1298	0.6039±0.0136	0.4125±0.0163	0.2155±0.0366	0.489±0.012	0.599±0.112
2	LTI_{79}	2.1915±0.0795	0.5826±0.0182	0.3946±0.0185	0.1989±0.0204	0.533±0.029	0.622±0.049
4	CMD-D	<u>1.7112</u>±0.1076	<u>0.5060</u>±0.0146	<u>0.2878</u>±0.0093	<u>0.0969</u>±0.0127	0.594±0.022	1.599±0.108
4	CMD-X	**1.5626**±0.0541	**0.4668**±0.0080	**0.2665**±0.0083	**0.0617**±0.0081	0.661±0.031	0.592±0.066
4	LTI_{64}	2.2446±0.0890	0.5933±0.0065	0.4057±0.0125	0.2062±0.0219	0.513±0.010	0.579±0.071
4	LTI_{79}	2.1963±0.0306	0.5812±0.0100	0.3935±0.0080	0.1948±0.0103	0.538±0.008	0.599±0.011
8	CMD-D	<u>1.6214</u>±0.0584	<u>0.4791</u>±0.0089	<u>0.2748</u>±0.0105	<u>0.0854</u>±0.0095	0.716±0.018	1.965±0.121
8	CMD-X	**1.5695**±0.0442	**0.4619**±0.0043	**0.2607**±0.0079	**0.0567**±0.0062	0.721±0.020	0.621±0.043
8	LTI_{64}	2.2050±0.0694	0.5857±0.0065	0.3974±0.0115	0.2017±0.0109	0.527±0.009	0.592±0.041
8	LTI_{79}	2.1748±0.0659	0.5825±0.0121	0.3894±0.0114	0.1936±0.0073	0.545±0.010	0.609±0.050
16	CMD-D	**1.5568**±0.0309	**0.4599**±0.0071	<u>0.2632</u>±0.0042	<u>0.0812</u>±0.0054	0.745±0.013	1.541±0.117
16	CMD-X	<u>1.5644</u>±0.0325	<u>0.4657</u>±0.0087	**0.2620**±0.0048	**0.0585**±0.0049	0.708±0.011	0.637±0.041
16	LTI_{64}	2.1741±0.0416	0.5866±0.0059	0.3940±0.0055	0.2100±0.0091	0.524±0.005	0.613±0.034
16	LTI_{79}	2.1724±0.0356	0.5864±0.0123	0.3914±0.0098	0.2020±0.0104	0.534±0.015	0.611±0.016
25	CMD-D	**1.5151**±0.0083	**0.4541**±0.0031	**0.2575**±0.0019	<u>0.0795</u>±0.0043	0.762±0.008	1.521±0.116
25	CMD-X	<u>1.5596</u>±0.0053	<u>0.4635</u>±0.0039	<u>0.2607</u>±0.0013	**0.0528**±0.0008	0.744±0.005	0.638±0.009
25	LTI_{64}	2.1628±0.0199	0.5845±0.0042	0.3918±0.0043	0.2092±0.0035	0.529±0.006	0.619±0.013
25	LTI_{79}	2.1629±0.0334	0.5864±0.0067	0.3913±0.0063	0.2078±0.0030	0.534±0.006	0.626±0.020

Table 8: **Physionet: complete component sweep.** Mean ± standard deviation across ten seeds (0–9); lower scores are better for error metrics (best results in bold, second-best results underlined). Coverage targets 0.80. Direct and LTI rows retain the first M codes of a 25-component head; CMD-D generates M components. LTI subscripts give the mode count K. Every score uses 25 paths.

M	Model	ES	VS	CRPS	PIT-ECE	Coverage	Width
1	CMD-D	2.9557±0.2028	6.9482±1.1193	0.7444±0.0484	0.1966±0.0137	0.098±0.047	0.263±0.134
1	CMD-X	**1.8001**±0.1316	**4.9372**±0.1123	**0.4248**±0.0412	**0.1499**±0.0505	0.437±0.152	0.726±0.253
1	LTI_{64}	2.3282±0.5044	5.1474±0.2015	0.5709±0.1494	0.2160±0.0891	0.326±0.081	0.694±0.187
1	LTI_{79}	1.9707±0.1524	5.0015±0.2632	0.4649±0.0469	0.1689±0.0705	0.425±0.126	0.787±0.232
2	CMD-D	2.1854±0.1056	5.6263±0.6165	0.5366±0.0261	**0.0916**±0.0075	0.427±0.030	1.078±0.125
2	CMD-X	**1.7935**±0.1699	**4.6352**±0.1197	**0.4034**±0.0507	0.1251±0.0696	0.539±0.037	0.921±0.189
2	LTI_{64}	2.0093±0.2750	4.8397±0.0810	0.4801±0.0834	0.1737±0.0855	0.456±0.101	0.887±0.180
2	LTI_{79}	1.9086±0.0968	4.9297±0.1841	0.4479±0.0253	0.1159±0.0456	0.574±0.182	1.236±0.553
4	CMD-D	1.8136±0.0640	4.9825±0.3766	0.4362±0.0167	**0.0379**±0.0096	0.669±0.023	1.696±0.148
4	CMD-X	**1.6676**±0.0810	**4.5179**±0.0896	**0.3677**±0.0239	0.0681±0.0535	0.648±0.076	1.079±0.237
4	LTI_{64}	1.7792±0.1375	4.7364±0.0690	0.4122±0.0398	0.1001±0.0600	0.621±0.098	1.154±0.295
4	LTI_{79}	1.7399±0.0441	4.7763±0.1033	0.4002±0.0124	0.0525±0.0276	0.715±0.100	1.509±0.288
8	CMD-D	1.6518±0.0466	4.6994±0.2757	0.3926±0.0126	**0.0281**±0.0113	0.764±0.023	1.979±0.169
8	CMD-X	**1.6219**±0.0495	**4.5888**±0.1062	**0.3568**±0.0130	0.0560±0.0395	0.713±0.073	1.223±0.222
8	LTI_{64}	1.7089±0.0586	4.7121±0.0621	0.3933±0.0155	0.0640±0.0397	0.749±0.079	1.517±0.203
8	LTI_{79}	1.7030±0.0692	4.7291±0.0893	0.3918±0.0191	0.0728±0.0414	0.694±0.072	1.274±0.233
16	CMD-D	**1.5341**±0.0335	**4.4931**±0.2028	0.3609±0.0094	**0.0312**±0.0087	0.790±0.025	1.885±0.159
16	CMD-X	1.5831±0.0081	4.5230±0.0672	**0.3462**±0.0029	0.0383±0.0100	0.786±0.047	1.463±0.197
16	LTI_{64}	1.6738±0.0312	4.7085±0.0669	0.3838±0.0103	0.0487±0.0299	0.757±0.073	1.506±0.177
16	LTI_{79}	1.6523±0.0360	4.7129±0.0626	0.3775±0.0112	0.0366±0.0130	0.775±0.050	1.521±0.165
25	CMD-D	**1.4763**±0.0286	**4.3916**±0.1764	0.3453±0.0082	0.0381±0.0099	0.820±0.026	1.932±0.164
25	CMD-X	1.5674±0.0066	4.5241±0.0345	**0.3422**±0.0019	0.0204±0.0031	0.790±0.005	1.464±0.035
25	LTI_{64}	1.6428±0.0166	4.7021±0.0199	0.3750±0.0050	0.0175±0.0015	0.788±0.005	1.572±0.032
25	LTI_{79}	1.6323±0.0216	4.7020±0.0573	0.3720±0.0063	**0.0160**±0.0024	0.784±0.006	1.530±0.058

Table 9: **BMS-air: complete component sweep.** Mean ± standard deviation across ten seeds (0–9); lower scores are better for error metrics (best results in bold, second-best results underlined). Coverage targets 0.80. Direct and LTI rows retain the first M codes of a 25-component head; CMD-D generates M components. LTI subscripts give the mode count K. Every score uses 25 paths.

M	Model	ES	VS	CRPS	PIT-ECE	Coverage	Width
1	CMD-D	16.511±0.843	**589.59**±88.43	0.5764±0.0930	**0.0713**±0.0311	0.362±0.062	0.664±0.221
1	CMD-X	**13.350**±0.863	726.41±97.27	**0.4982**±0.1349	0.1190±0.0326	0.531±0.033	1.427±0.261
1	LTI_{64}	15.047±1.007	848.60±79.30	0.6463±0.0768	0.1097±0.0539	0.523±0.104	1.687±0.436
1	LTI_{74}	15.023±0.964	801.73±45.37	0.6555±0.0930	0.1109±0.0317	0.525±0.086	1.739±0.028
2	CMD-D	15.135±0.623	**526.78**±66.59	0.4781±0.0855	**0.0533**±0.0310	0.499±0.046	1.203±0.226
2	CMD-X	**12.336**±0.866	703.74±47.37	**0.4574**±0.0967	0.0657±0.0423	0.611±0.095	1.582±0.180
2	LTI_{64}	14.669±0.671	855.01±6.26	0.6246±0.0561	0.1041±0.0432	0.552±0.056	1.793±0.213
2	LTI_{74}	14.760±0.682	742.21±33.68	0.6474±0.0752	0.1019±0.0303	0.540±0.071	1.704±0.069
4	CMD-D	14.707±0.419	**502.23**±42.41	0.4338±0.0489	**0.0381**±0.0164	0.650±0.032	1.547±0.215
4	CMD-X	**12.018**±0.452	669.92±42.83	**0.4302**±0.0540	0.0582±0.0215	0.635±0.076	1.566±0.170
4	LTI_{64}	14.466±0.548	837.09±18.35	0.6127±0.0391	0.0993±0.0287	0.557±0.042	1.755±0.201
4	LTI_{74}	14.638±0.561	738.43±33.78	0.6296±0.0582	0.0947±0.0260	0.547±0.058	1.698±0.044
8	CMD-D	12.772±0.283	**493.50**±26.52	0.4026±0.0159	**0.0276**±0.0084	0.714±0.026	1.678±0.200
8	CMD-X	**11.873**±0.318	672.01±53.23	**0.3969**±0.0118	0.0439±0.0212	0.655±0.054	1.600±0.209
8	LTI_{64}	14.322±0.623	842.48±12.12	0.6003±0.0348	0.0923±0.0243	0.576±0.060	1.778±0.191
8	LTI_{74}	14.539±0.500	740.53±46.15	0.6164±0.0425	0.0893±0.0240	0.557±0.065	1.712±0.056
16	CMD-D	11.906±0.153	**481.64**±21.95	0.3970±0.0081	**0.0212**±0.0085	0.743±0.028	1.604±0.198
16	CMD-X	**11.557**±0.035	661.54±72.07	**0.3843**±0.0036	0.0339±0.0065	0.677±0.024	1.614±0.078
16	LTI_{64}	14.279±0.679	856.98±33.49	0.5992±0.0397	0.0894±0.0277	0.585±0.069	1.802±0.100
16	LTI_{74}	14.445±0.435	746.82±44.18	0.6105±0.0331	0.0882±0.0170	0.569±0.058	1.768±0.064
25	CMD-D	**11.246**±0.106	**469.64**±18.63	0.3846±0.0048	**0.0182**±0.0066	0.748±0.012	1.602±0.192
25	CMD-X	11.613±0.135	659.12±57.60	**0.3823**±0.0018	0.0337±0.0069	0.673±0.022	1.591±0.029
25	LTI_{64}	14.259±0.665	861.08±42.51	0.5964±0.0380	0.0889±0.0271	0.589±0.072	1.818±0.102
25	LTI_{74}	14.402±0.357	756.30±22.78	0.6045±0.0239	0.0863±0.0140	0.577±0.050	1.803±0.032

Table 10: **NOAA-UK: additional metrics and controls.** Means over ten seeds (0–9); ± denotes standard deviation. Lower scores are better for error metrics (best results in bold, second-best results underlined). Coverage targets 0.80. The last column gives mean coverage/width.

Method	ES	VS	CRPS	PIT-ECE	Cov./width
CMD-D(25)	**13.608**±0.159	**438.8**±8.1	**0.4503**±0.0035	0.0943±0.0308	0.731/1.940
Gaussian collapse	15.317±0.161	466.4±10.2	0.4987±0.0057	0.1442±0.0324	0.712/1.846
Means as paths	15.719±0.216	570.1±21.6	0.5055±0.0069	0.2768±0.0676	0.630/1.491
Block-diagonal K_j	14.354±0.224	481.9±13.7	**0.4503**±0.0035	0.0943±0.0309	0.731/1.941
Chirp-MSE diffusion	14.504±0.171	490.9±23.8	0.5024±0.0136	0.1047±0.0694	0.615/1.447
QuITE++ + DG	19.300±1.541	918.1±79.0	0.6641±0.0644	0.2380±0.0285	0.394/1.414
sdevi	20.207±0.241	630.9±3.5	0.7089±0.0103	0.2607±0.0077	0.235/0.938
TFMixer + DG	14.409±0.025	1165.2±19.8	0.5485±0.0009	**0.0363**±0.0047	0.736/1.676
LLapDiff	16.276±0.102	811.01±59.1	0.5491±0.0242	0.1341±0.0088	0.535/1.603
diagonal Gaussian	23.832±0.383	724.9±25.9	0.8367±0.0157	0.4288±0.0034	0.167/1.259
low-rank Gaussian	24.038±1.026	660.8±78.0	0.8498±0.0402	0.4303±0.0061	0.156/1.169
Student-t	22.282±0.320	718.3±34.0	0.8161±0.0157	0.4121±0.0085	0.209/1.286
MOSES	22.895±2.878	719.8±23.8	0.8205±0.0443	0.4052±0.0303	0.243/1.646
ProFITi	16.172±1.151	902.9±35.6	0.5124±0.0494	0.3306±0.0232	0.516/1.718
CRU	21.300±4.938	770.0±159.1	0.7450±0.1159	0.3917±0.0423	0.356/1.490
NCDSSM	24.395±0.245	820.8±31.2	0.8542±0.0111	0.4241±0.0047	0.185/1.444
Spectral-mixture GP	25.022±0.048	480.2±19.8	0.8851±0.0020	0.4383±0.0069	0.110/1.107
PatchTST + Gaussian	21.972±3.852	587.3±56.0	0.6575±0.0045	0.1786±0.0190	0.435/1.673

Table 11: **USHCN: additional metrics and controls.** Means over ten seeds (0–9); ± denotes standard deviation. Lower scores are better for error metrics (best results in bold, second-best results underlined). Coverage targets 0.80. The last column gives mean coverage/width.

Method	ES	VS	CRPS	PIT-ECE	Cov./width
CMD-D(25)	**1.5151**±0.0083	**0.4541**±0.0031	**0.2575**±0.0019	0.0795±0.0043	0.762/1.521
Gaussian collapse	1.6640±0.0095	0.6015±0.0122	0.3347±0.0088	0.1337±0.0271	0.627/1.542
Means as paths	1.6322±0.0052	0.5798±0.0075	0.2764±0.0024	0.1224±0.0736	0.619/1.371
Block-diagonal K_j	1.5243±0.0080	0.4909±0.0042	**0.2575**±0.0018	0.0795±0.0043	0.762/1.521
Chirp-MSE diffusion	1.7468±0.0105	0.5053±0.0082	0.2770±0.0036	0.1099±0.0556	0.664/1.439
QuITE++ + DG	1.6859±0.0422	3.9283±0.2631	0.3239±0.0129	0.0820±0.0077	0.857/2.562
sdevi	1.6559±0.0448	2.5223±0.0081	0.2989±0.0100	**0.0323**±0.0017	0.742/1.866
TFMixer + DG	1.9423±0.1894	5.4755±0.5691	0.6148±0.1180	0.1343±0.0281	0.751/2.915
LLapDiff	1.9590±0.0152	0.5544±0.0037	0.3873±0.0081	0.1038±0.0022	0.542/0.843
diagonal Gaussian	2.2069±0.0456	0.9458±0.0652	0.4135±0.0147	0.1652±0.0128	0.628/1.590
low-rank Gaussian	2.2188±0.1050	0.9391±0.0341	0.4186±0.0133	0.1749±0.0871	0.698/0.839
Student-t	2.2295±0.0366	0.8079±0.0827	0.3569±0.0045	0.1399±0.0058	0.680/0.651
MOSES	1.8201±0.1372	2.6614±0.3565	0.3846±0.0326	0.0878±0.0207	0.709/2.208
ProFITi	1.7014±0.0731	2.1685±0.5514	0.3643±0.0141	0.0867±0.0165	0.692/1.255
CRU	1.9968±0.0150	3.9562±0.1526	0.4811±0.0726	0.1309±0.0243	0.858/2.233
NCDSSM	2.2467±0.0246	2.4797±0.1827	0.3273±0.0054	0.1148±0.0054	0.723/1.810
Spectral-mixture GP	2.3888±0.1461	3.7202±0.6385	0.4089±0.0155	0.1116±0.0039	0.718/2.602
PatchTST + Gaussian	2.3771±0.2802	4.1979±0.3449	0.4054±0.0696	0.1142±0.0137	0.719/3.472

Table 12: **Physionet: additional metrics and controls.** Means over ten seeds (0–9); ± denotes standard deviation. Lower scores are better for error metrics (best results in bold, second-best results underlined). Coverage targets 0.80. The last column gives mean coverage/width.

Method	ES	VS	CRPS	PIT-ECE	Cov./width
CMD-D(25)	**1.4763**±0.0286	**4.3916**±0.1764	**0.3453**±0.0082	0.0381±0.0099	0.820/1.932
Gaussian collapse	1.5492±0.0240	4.7505±0.1480	0.3607±0.0077	0.0295±0.0089	0.773/1.498
Means as paths	1.5277±0.0202	4.5967±0.1223	0.3617±0.0090	**0.0104**±0.0054	0.705/1.199
Block-diagonal K_j	<u>1.4816</u>±0.0234	4.6243±0.1527	<u>0.3454</u>±0.0082	0.0381±0.0099	0.820/1.933
Chirp-MSE diffusion	1.5294±0.0274	<u>4.5394</u>±0.1588	0.3621±0.0121	<u>0.0191</u>±0.0056	0.756/1.297
QuITE++ + DG	2.2912±0.0027	6.2767±0.1705	0.3828±0.0033	0.0203±0.0023	0.728/1.537
sdevi	2.3040±0.1025	5.3495±0.4615	0.3860±0.0021	<u>0.0191</u>±0.0129	0.676/1.149
TFMixer + DG	2.4093±0.0314	10.1333±0.9744	0.4065±0.0153	0.0277±0.0028	0.848/2.082
LLapDiff	1.6389±0.0167	4.8397±0.0390	0.3760±0.0061	0.0346±0.0079	0.786/1.780
diagonal Gaussian	2.3711±0.0202	9.8000±0.2994	0.4264±0.0105	0.0842±0.0216	0.703/1.194
low-rank Gaussian	2.3943±0.0626	9.6762±0.3814	0.4211±0.0198	0.0308±0.0158	0.811/2.037
Student-t	2.3369±0.0114	9.9677±0.0849	0.4147±0.0066	0.0272±0.0034	0.721/1.231
MOSES	2.2532±0.1426	10.2930±0.2996	0.4134±0.0160	0.0842±0.0145	0.751/1.755
ProFITi	1.7870±0.0188	6.1533±0.1162	0.3755±0.0047	0.0556±0.0117	0.754/1.874
CRU	2.5593±0.0250	11.8286±1.0841	0.4343±0.0057	0.0393±0.0130	0.777/2.128
NCDSSM	2.4421±0.0242	7.7117±0.0196	0.3918±0.0073	0.0704±0.0189	0.758/1.786
Spectral-mixture GP	2.2737±0.0065	6.3185±0.0324	0.3916±0.0041	0.0229±0.0026	0.795/1.584
PatchTST + Gaussian	2.3912±0.0221	9.7154±1.0125	0.4567±0.0210	0.0365±0.0077	0.793/2.068

Table 13: **BMS-air: additional metrics and controls.** Means over ten seeds (0–9); ± denotes standard deviation. Lower scores are better for error metrics (best results in bold, second-best results underlined). Coverage targets 0.80. The last column gives mean coverage/width.

Method	ES	VS	CRPS	PIT-ECE	Cov./width
CMD-D(25)	**11.246**±0.106	**469.64**±18.63	**0.3846**±0.0048	**0.0182**±0.0066	0.748/1.602
Gaussian collapse	13.453±0.117	514.43±25.80	0.4250±0.0058	<u>0.0328</u>±0.0089	0.629/1.561
Means as paths	14.689±0.149	553.25±30.44	0.4263±0.0051	0.0791±0.0063	0.616/1.084
Block-diagonal K_j	<u>11.732</u>±0.121	507.54±20.54	<u>0.3847</u>±0.0047	**0.0182**±0.0066	0.748/1.603
Chirp-MSE diffusion	13.752±0.115	<u>485.74</u>±20.29	0.4034±0.0057	0.0699±0.0082	0.650/1.098
QuITE++ + DG	12.922±0.047	1213.26±22.09	0.5277±0.0032	0.0798±0.0049	0.870/2.937
sdevi	13.986±0.285	627.32±3.43	0.5884±0.0172	0.1639±0.0092	0.371/1.000
TFMixer + DG	13.093±0.179	795.12±16.86	0.5107±0.0063	0.0772±0.0155	0.684/1.522
LLapDiff	13.211±0.397	710.38±52.67	0.5939±0.0274	0.0340±0.0047	0.648/1.976
diagonal Gaussian	17.701±0.226	569.96±33.05	0.5386±0.0105	0.1819±0.0162	0.581/1.321
low-rank Gaussian	20.112±0.207	665.32±86.97	0.5747±0.0130	0.1939±0.0127	0.678/1.935
Student-t	19.635±0.192	617.48±15.65	0.5377±0.0129	0.1471±0.0167	0.746/2.080
MOSES	17.252±3.523	592.79±93.23	0.5505±0.0741	0.1762±0.0671	0.799/3.858
ProFITi	15.112±0.661	864.76±45.74	0.5891±0.0610	0.1862±0.0363	0.742/2.018
CRU	21.621±1.525	724.21±76.68	0.7320±0.0315	0.2316±0.0483	0.737/2.245
NCDSSM	17.973±0.157	613.04±38.89	0.5793±0.0096	0.2877±0.0043	0.661/2.109
Spectral-mixture GP	19.379±0.104	652.57±8.17	0.6931±0.0090	0.2380±0.0045	0.685/0.933
PatchTST + Gaussian	16.321±0.738	557.29±67.21	0.6203±0.0606	0.1356±0.0596	0.643/1.450

F ADDITIONAL PREDICTION COMPARISONS

Single-component direct prediction. The $M_{\text{pool}} = 1$ CMD-X row retains the first code from a mixture-trained head. It leverages query-grid variance propagation and correlated path sampling. It is not a separately trained one-component model: the training objective remains the 25-component mixture likelihood.

Small-pool operating points. We compared retained pools of eight and 25 components using the same 25-path scoring budget (Table 14). At pool size eight, CMD-X achieved lower mean ES on the three environmental datasets shown and lower mean VS on USHCN. The allocation and learned-code prefix conventions are formalized in Appendix D; the generator-call counts are distinct from the measured latencies in Appendix I.

Table 14: **A compact operating-point comparison.** ES / VS; all scores use 25 paths. Direct rows retain components from a 25-code checkpoint. The final column gives trajectory-level denoiser evaluations for CMD-D and direct passes for CMD-X. CMD-D has 64 sequential steps and can batch trajectories into 64 network invocations. The table covers the three environmental datasets.

Model	NOAA-UK	USHCN	BMS-air	Generation work
CMD-X(8)	13.698 / 520.93	1.5695 / 0.4619	11.873 / 672.01	1 direct pass
CMD-D(8)	14.287 / 459.20	1.6214 / 0.4791	12.772 / 493.50	512 evaluations
CMD-X(25)	13.771 / 518.61	1.5596 / 0.4635	11.613 / 659.12	1 direct pass
CMD-D(25)	13.608 / 438.80	1.5151 / 0.4541	11.246 / 469.64	1,600 evaluations

Mean-synthesizer comparison. We assessed four mean synthesizers under MSE training on USHCN, including re-gridded selective diagonal and damped oscillatory SSM controls (Table 15). Related SSM designs incorporate input-dependent selection in Mamba (Gu & Dao, 2023), forced harmonic oscillators in LinOSS (Rusch & Rus, 2025), and learned damping in D-LinOSS (Boyer et al., 2025). We attained the lowest mean MSE and CRPS with the chirp-modal synthesizer. This comparison evaluates mean parameterizations under the MSE objective and complements the full-grid mixture-likelihood comparisons in the main paper.

Re-gridding protocol. Both SSM controls operate on a uniform grid with spacing $\Delta = 1$ day, the native resolution of the USHCN records, anchored at each window's forecast origin. For positive query offsets $t_1, \ldots, t_h$, the corresponding readout cells and rollout length are

$$k_r = \max\{1, \operatorname{round}(t_r/\Delta)\}, \qquad N = \max_{r=1,\ldots,h} k_r.$$

The recurrence visits cells $k = 1, \ldots, N$, with each cell receiving a token derived from the shared frozen history summary and a cell-position embedding. No future observations are exposed to the recurrence, and no interpolation or imputation is executed over the forecast horizon. Predictions are read out at the original query cells, and the MSE loss is computed only at those cells; intermediate cells advance the recurrent state without an additional loss term. The history encoder, forecast origin, query targets, training objective, optimizer, and training budget are preserved across the compared cores.

Query alignment and rollout length. All audited query offsets lie exactly on the daily grid, with zero mean and maximum temporal displacement. Across the audited windows, the median rollout length is 53 cells, with a 95th percentile of 82, a 99th percentile of 102, and a maximum of 534. Thus, the daily grid imposes neither query-time displacement nor horizon-target imputation in the audited windows. The SSM controls nevertheless propagate through intervening daily cells, whereas the modal cores evaluate directly at the requested offsets; the comparison includes this difference in how the mean trajectory is constructed.

Table 15: **Mean-synthesizer comparison on USHCN.** MSE-trained models; mean ± standard deviation. Lower is better. Both SSM controls use the daily-grid protocol described in the text, with exact query alignment and no horizon imputation. The fixed-pole control uses the same K as the chirp-modal model.

Mean synthesizer	MSE	CRPS
Chirp-modal	**0.291** $\pm$ 0.035	**0.276** $\pm$ 0.013
Fixed-pole (equal K)	0.348 $\pm$ 0.057	0.328 $\pm$ 0.020
Selective diagonal SSM (re-gridded)	0.566 $\pm$ 0.075	0.409 $\pm$ 0.052
Damped oscillatory SSM (re-gridded)	0.777 $\pm$ 0.179	0.593 $\pm$ 0.093

G MARGINAL-PRESERVING COVARIANCE INTERVENTION

Proposition 3 (Removing residual temporal dependence)**.** *For any finite Gaussian mixture on a query grid, replace each component covariance by*

$$K_j^{\mathrm{time-ind}} = \mathrm{blockdiag}_{r=1}^{h} K_j(t_r, t_r),$$

retaining its mean, weight, and one global component identity across all times. Every per-time mixture distribution is invariant. A decoder that operates independently at each time preserves equality of per-time observed distributions.

Proof. Each diagonal time block and each mean restriction is the same in the original and intervened Gaussian, hence their per-time Gaussian marginals coincide. The same mixture weights yield the same per-time mixture marginal. Applying the same measurable per-time decoder to equal latent laws preserves equality. Off-diagonal covariance inside components is removed, while covariance from the shared component mean remains: $\mathrm{Cov}(z_r, z_s) = \mathrm{Cov}_j(m_j(t_r), m_j(t_s))$ for $r \neq s$ after intervention. □

Real-data intervention. We applied this transformation to the same CMD-D checkpoint and 25 inferred components employed for the full-mixture forecast, then evaluated 25 decoded paths. Figure 4 illustrates the score changes on all four datasets. The underlying per-time laws are preserved; finite-ensemble marginal scores can still differ. Appendix Tables 10–13 report means and seed standard deviations for this intervention.

Separating the sources of dependence. Intuitively, the intervention leaves the uncertainty at each query fixed but modifies how deviations around a component mean co-occur across queries (Figure 4). A single component identity still links the times through their component means. Gaussian moment matching instead preserves the full covariance and changes the distributional shape; using component means as paths removes all residual variation. These algebraic distinctions identify the aspects of a predictive mixture targeted by each control.

H SYNTHETIC CONSTRUCTION: LEARNING A JOINT FORECAST

H.1 ORACLE AND CONTROLLED DISTRIBUTIONS

We designed a two-regime forecasting law to study shared regime identity and temporal uncertainty, then fitted CMD-X to independent paths from that law. We fixed the forecast cutoff at zero and specified $h = 10$ irregular query offsets,

$$(t_1, \ldots, t_{10}) = (0.7, 1.5, 2.2, 3.6, 4.1, 5.4, 6.3, 6.9, 8.2, 9.4). \tag{27}$$

The two regime means and their shared residual covariance are

$$m_b(t) = \cos\left(\frac{2\pi t}{5.5} + (2b-1)\frac{0.06t^2}{2}\right), \qquad b \in \{0, 1\}, \tag{28}$$

$$K_\epsilon(t, s) = 0.35^2 \left[e^{-|t-s|/3} - e^{-(t+s)/3}\right]. \tag{29}$$

Both means leave the cutoff together; regime 0 slows down and regime 1 speeds up. The residual is an Ornstein–Uhlenbeck process started at zero, with positive-definite covariance on the strictly positive query grid. Writing $m_b = (m_b(t_r))_{r=1}^{10}$ and $K_\epsilon = (K_\epsilon(t_r, t_s))_{r,s=1}^{10}$, we draw $B \sim \mathrm{Bernoulli}(1/2)$ once per path, independently of $\epsilon \sim \mathcal{N}(0, K_\epsilon)$:

$$z = m_B + \epsilon, \qquad P = \tfrac{1}{2}\mathcal{N}(m_0, K_\epsilon) + \tfrac{1}{2}\mathcal{N}(m_1, K_\epsilon). \tag{30}$$

Preserving marginals while changing regime identity. For P_{sw}, we draw independent $B_r \sim \mathrm{Bernoulli}(1/2)$ at each query and set $z_r = m_{B_r}(t_r) + \epsilon_r$, keeping the same joint residual law. Every one-time marginal therefore equals P_r exactly. Its joint density is

$$p_{\mathrm{sw}}(z) = 2^{-h} \sum_{\beta \in \{0,1\}^h} \mathcal{N}(z; m_\beta, K_\epsilon), \qquad (m_\beta)_r = m_{\beta_r}(t_r). \tag{31}$$

We compute the sum over all 2^{10} branch assignments. We also define $P_{\mathrm{ind}} = \prod_r P_r$, which independently samples the exact one-time marginals and eliminates both shared regime identity and residual temporal dependence.

Preserving the full mean and covariance. With $\bar{m} = (m_0 + m_1)/2$ and $\delta = m_1 - m_0$, we use

$$P_{\mathrm{MM}} = \mathcal{N}(\bar{m}, K_\epsilon + \delta\delta^\top/4). \tag{32}$$

This is the $M = 2$ instance of (14). It preserves the complete mean vector and covariance matrix of P, including the between-regime covariance.

Probability between regimes. We project the two-time vector $x = (z(t_5), z(t_8))^\top$, at times 4.1 and 6.9, onto the line joining its regime means μ_0 and μ_1. For $v = \mu_1 - \mu_0$, define

$$s(x) = \frac{v^\top[x - (\mu_0 + \mu_1)/2]}{\|v\|^2}. \tag{33}$$

The regime means map to $-1/2$ and $1/2$. The shaded region in Figure 3c is $|s| < 1/4$, the middle half of that segment. Integrating the projected Gaussian densities gives probability 0.02909 under P and 0.37122 under P_{MM}.

H.2 Learning the joint law with CMD-X

Model. We employ CMD-X's shared generator $G_\theta(E, e_j)$, learned component codes, bounded chirp pole fields in (20), positive variance recurrence in (22), and full-grid mixture NLL in (12). We specify a scalar latent ($d_z = 1$), identity representation maps, and the same constant context $E = 1$ for every path. This isolates learning a joint distribution with the component family and its generator. The head has width 256 and depth 2, with $M_{\mathrm{train}} = M_{\mathrm{pool}} = 25$ uniform components, $K = 128$ modes, $\alpha = 1$, four harmonic basis functions, and time scale $L = 10$. We use the original query offsets, lower decay bound 10^{-3}. Oracle information enters training through unlabeled paths alone; the model observes neither regime labels nor oracle parameters.

We retain the same component and likelihood modules as in the forecasting experiments. Since the context is fixed, we evaluate the head once per batch and broadcast its components over targets, preserving the per-example loss and gradients. Training uses float32 and evaluation float64. For component j, the likelihood uses relative diagonal jitter $\eta_j = 10^{-4} h^{-1} \operatorname{tr}(K_j)$. We inject independent noise with variance η_j to modal path draws so that both sampling and density evaluation share $K_j + \eta_j I$.

Data and optimization. We generated 20,000 training paths, 4,000 validation paths, and 20 independent test batches of 1,000 paths using separate random streams. Training uses AdamW with weight decay 10^{-4}, batch size 256, gradient clipping at 1, and cosine annealing over 200 epochs. We initialized component codes from $\mathcal{N}(0, 0.02^2)$, the readout from $\mathcal{N}(0, 0.01^2)$, and chirp coefficients with standard deviation 10^{-4}. A validation pilot on seeds 0–2 compared learning rates 10^{-4}, 3×10^{-4}, and 10^{-3}; 3×10^{-4} had the lowest mean best-validation joint NLL and was selected. We trained seeds 0–9 with this setting and retained each seed's best-validation checkpoint. All ten runs completed and entered the evaluation.

Paired likelihood evaluation. We compare the exact finite-mixture density Q_X with its same-checkpoint Gaussian collapse $Q_{X,\mathrm{MM}}$, formulated by (14), and $Q_{X,\mathrm{diag}}$, which diagonalizes each

component covariance while retaining its mean, weight, and one component identity per path. The latter maintains all learned one-time marginals (Proposition 3). On each shared batch of $n = 1,000$ oracle paths we compute

$$\widehat{\Delta}(q) = \frac{1}{nh}\sum_{i=1}^{n}[\log p(z^{(i)}) - \log q(z^{(i)})], \qquad z^{(i)} \sim P. \tag{34}$$

The marginal counterpart substitutes each joint log density by the sum of its one-time log densities. We first average the 20 batch scores within a training seed, then report means and standard deviations over ten seeds. Paired 95% t-intervals employ the ten seed-level score differences on the shared test set. For fixed oracle controls, 95% t-intervals instead use the 20 independent test batches.

Threshold and path event. We fixed $u = 0.75265295$ at the pooled 80th percentile of query values in a separate sample of 2,000 oracle paths, before final evaluation. We measure

$$A_u = \{z : \text{there exists } r \in \{1, \dots, 9\} \text{ with } z_r > u \text{ and } z_{r+1} > u\}. \tag{35}$$

For learned laws, we employ 200,000 independent full-path draws per seed (Monte Carlo standard error at most 0.0011). For P, P_{sw}, and P_{MM}, we use two million draws per law (standard error at most 0.00036); for P_{ind}, an exact consecutive-exceedance recursion is applied. The observed test frequency is 0.4670, with 95% interval $[0.4604, 0.4736]$ across batches; the oracle probability is 0.4642. Projected densities and probabilities, including $Q(|s| < 1/4)$ and $Q(s < 0)$, are computed analytically from the Gaussian mixture projections.

H.3 Supplementary diagnostics and results

Figure 5 illustrates the learned distributions and joint diagnostics across ten training seeds. Table 16 details their joint and marginal likelihoods, and Table 17 presents the measured path-event and between-regime probabilities.

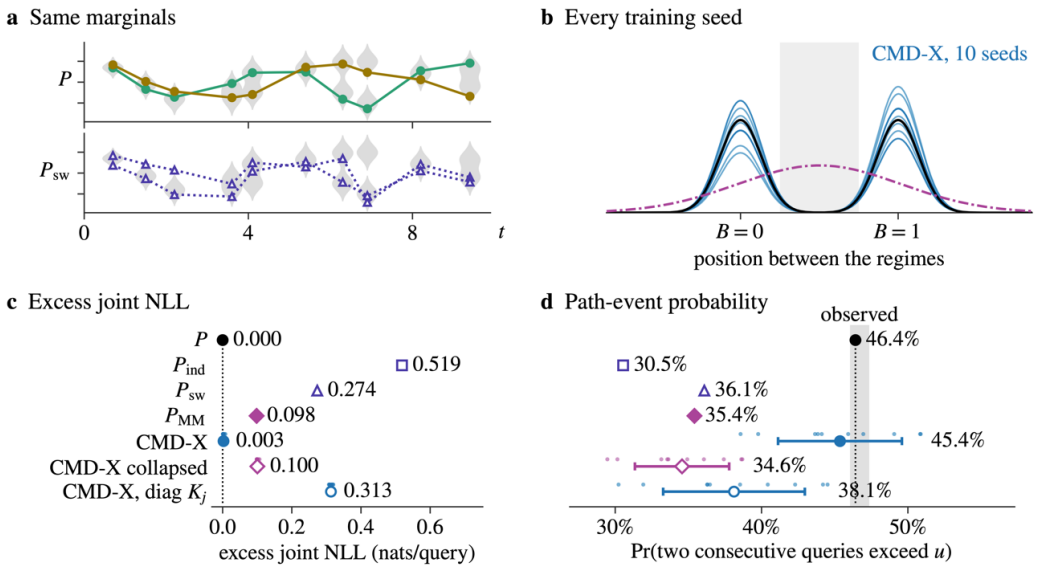


Figure 5: **Learned joint structure across ten training seeds.** (a) Oracle and branch-switching paths share the grey one-time marginals. (b) Projected densities for all CMD-X seeds (blue), the oracle (black), and its Gaussian collapse (purple); shading marks $|s| < 1/4$. (c–d) Excess joint NLL and path-event probabilities, including same-checkpoint controls. Small points denote training seeds; large learned-model points and bars show means ± 1 seed standard deviation. The grey band in (d) is the 95% interval for observed frequency across 20 independent test batches.

Table 16: **Learned likelihoods and same-checkpoint controls.** Excess NLL in nats/query relative to P, mean $\pm$ standard deviation over ten training seeds. The $M = 2$ row is a separately trained diagnostic.

Forecast	Joint NLL	Marginal NLL
CMD-X, $M = 25$	0.00255 ± 0.00226	0.00786 ± 0.00945
Gaussian collapse	0.10040 ± 0.00253	0.07776 ± 0.00911
Diagonal covariance	0.31314 ± 0.00455	0.00786 ± 0.00945
CMD-X, $M = 2$	0.00074 ± 0.00011	0.000069 ± 0.000013

Learning dependence and distributional shape. Across ten seeds, CMD-X achieves excess joint NLL 0.00255 ± 0.00226 nats/query. Gaussian collapse increases it by 0.09785 (paired 95% interval $[0.09765, 0.09806]$), while the diagonal intervention increases it by 0.31059 $[0.30824, 0.31293]$ despite preserving every learned marginal. The model therefore learns both temporal residual dependence and mixture structure beyond the full mean and covariance. On the same test paths, P_{ind}, P_{sw}, and P_{MM} yield excess joint NLL 0.519 $[0.515, 0.522]$, 0.274 $[0.272, 0.275]$, and 0.098 $[0.097, 0.099]$, respectively. The first two attain exactly zero marginal excess NLL; the Gaussian records 0.070 $[0.069, 0.072]$.

Table 17: **Path-event and between-regime probabilities.** Learned rows report means $\pm$ standard deviations across ten training seeds. Oracle controls are fixed laws.

Forecast	$\Pr(A_u)$	$\Pr(\lvert s\rvert < 1/4)$
P	0.4642	0.02909
P_{ind}	0.3054	0.20648
P_{sw}	0.3609	0.18021
P_{MM}	0.3541	0.37122
CMD-X, $M = 25$	0.4536 ± 0.0423	0.02835 ± 0.00025
Gaussian collapse	0.3457 ± 0.0322	0.37070 ± 0.00058
Diagonal covariance	0.3812 ± 0.0484	0.05517 ± 0.00027
CMD-X, $M = 2$	0.4665 ± 0.0010	0.02858 ± 0.00013

Regime allocation and event probabilities. Every seed captures both regimes, with 8–15 of the 25 component means on the regime-0 side of the projection. The probability between regimes is 0.02835 ± 0.00025, proximal to the oracle's 0.02909. Event probability is close on average (0.4536 ± 0.0423 versus 0.4642) and varies across seeds. To examine this variation, we group components by the sign of their projected means and estimate event probabilities within each group from the same draws. These span from 0.6781 to 0.6868 on the regime-0 side and 0.2366 to 0.2506 on the regime-1 side; the oracle regime probabilities are 0.6794 and 0.2486. Giving the two learned groups equal total weight yields probabilities 0.4597–0.4674. This diagnostic pinpoints imbalance in the allocation of uniform components as the main contributor to event variation, alongside smaller within-regime discrepancies.

Two-component diagnostic. We separately trained the same head with $M_{\mathrm{train}} = M_{\mathrm{pool}} = 2$, retaining the other settings and the ten-seed protocol. Every run assigns one component to each regime. Its excess joint NLL is 0.00074 ± 0.00011 and its event probability is 0.4665 ± 0.0010. This diagnostic demonstrates that a two-component head can reconstruct the balanced regime allocation; the primary experiment employs $M = 25$ throughout.

Displayed paths. Panels (b–c) of Figure 3 leverage seed 0's best-validation checkpoint and the first six draws of a fixed display stream. Panel (d) summarizes all ten seeds. The common illustrated history is a period-5.5 cosine with Gaussian noise of standard deviation 0.08. Data generation, optimization randomness, event estimation, and displayed paths employ separate random streams.

I COMPUTATIONAL COMPLEXITY

Component evaluation. With fixed harmonic width, evaluating means, marginal variances, or one joint latent path demands $O(Khd_z)$ per component. We construct the d_z temporal covariance blocks in $O(Kh^2d_z)$ operations and store $O(h^2d_z)$ values. Full-grid GP likelihood evaluation necessitates a Cholesky factorization of each block, costing $O(d_zh^3)$ per component. Modal Markov sampling leverages the analytic transitions and scalar innovation variances directly.

Training. The direct mixture objective evaluates all M_{train} component densities for each training example. The denoising objective performs one network inference at a sampled diffusion level, followed by the component's full-grid likelihood and backpropagation. The reverse trajectory is used during generation. The history encoder and VAE remain frozen during stage-2 training.

Generation. CMD-D employs $64 \times M$ trajectory-level denoiser evaluations to generate M components: each trajectory has 64 sequential reverse steps. At $M = 25$, this totals 1,600 evaluations, or 64 network invocations when all trajectories are batched together. CMD-X generates all 25 learned-code components in one direct pass, leveraged by each retained-prefix sweep. Table 14 reports this work accounting. Equal trajectory-level counts do not imply equal FLOPs, latency, or training compute.

Fixed-anchor evaluation. For a fixed anchor grid with A intervals, the construction evaluates the pole fields at h query nodes and $A + 1$ anchors, demanding $O(K(h + A + 1)d_z)$ for means and scalar variances at fixed harmonic width. Its monotone energy guarantees nonnegative innovation variances between arbitrary sorted queries, enabling the same modal Markov sampling scheme. To recover observation-space paths, we additionally deploy the shared per-time VAE decoder.

Empirical runtime benchmarking. For the three environmental datasets, we calculated the end-to-end inference latency for generating 25 decoded observation-space paths. Timings were logged on a single NVIDIA RTX A6000 (48 GB) in fp32 with TF32 tensor cores enabled. We computed the median of 120 trials using asynchronous CUDA events (`torch.cuda.Event`) after 20 warm-up passes to mitigate cold-start anomalies. Data loading and host-to-device copies were omitted from these timings.

Figure 6 illustrates the empirical latencies. At batch size one, CMD-X requires 19–20 ms, batched CMD-D 128–151 ms, and sequential CMD-D 2,601–2,653 ms. Row-wise iterative/direct ratios are 6.7–7.6× for batched trajectories and 130–140× for sequential trajectories. Across both displayed batch sizes, these ranges span 6.7–20.2× and 24.7–139.6×, respectively. Batch-size-32 entries denote latencies for the entire batch, not per-window times.

J RELATION TO LLAPDIFF AND INTERPRETATION OF UNCERTAINTY

Fixed-pole correspondence. LLapDiff incorporates a stable complex-conjugate-pole modal synthesizer inside a latent denoiser (You et al., 2026). Its poles depend on context and the noisy latent, and remain constant across physical time within each synthesizer evaluation. Setting the chirp coefficients to zero in (6) recovers this modal sum before its output transformation. We augment the modes with positive initial variance and process-noise rates; our direct fixed-pole controls adopt the same Gaussian construction and full-grid mixture objective as CMD-X.

Output transformation. The multiplicative output scale α is already included in (6) and (8): its effect is linear on the mean and quadratic on the covariance. No further scaling is applied to those formulas. The nonlinear observation decoder maps latent path samples into the observed channels. Accordingly, the Gaussian density and exact moment-matching statements concern the latent representation, while the forecasting scores evaluate decoded samples.

Distributional distinctions. The fixed-pole and chirp direct heads share conditioning, learned-code generation, and the full-grid mixture objective. Chirp-MSE instead supplies an ensemble from an MSE-trained diffusion model with the chirp-modal mean backbone. The within-component and between-component uncertainties are defined in Section 2.1; the interventions in Section 3.3 modify them separately. Positive query-grid propagation induces the reported finite Gaussian laws. Appendix B.1 establishes the history-only, fixed-anchor conditions for consistency across query sets.

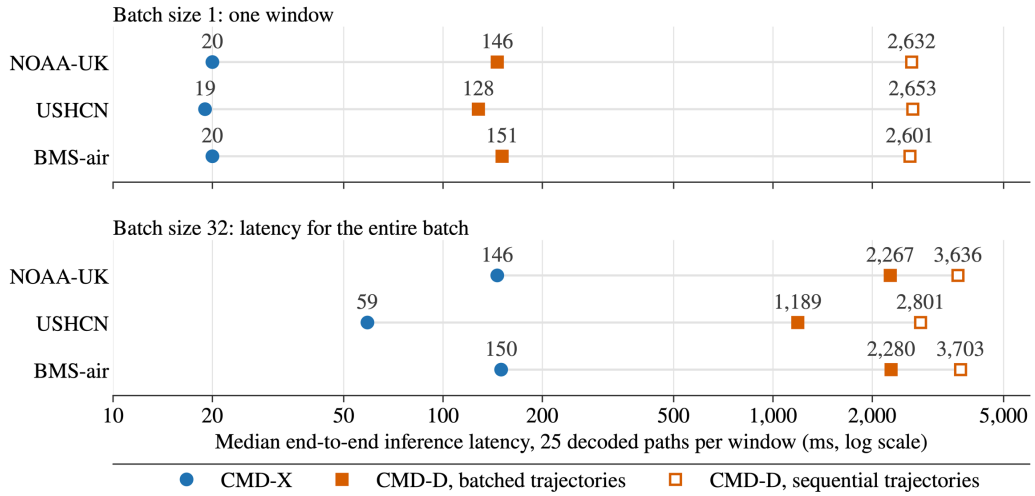


Figure 6: **Measured generation latency.** Median end-to-end inference latency for 25 decoded paths per window on an NVIDIA RTX A6000. Batch-size-32 values cover the entire batch. CMD-D trajectories are evaluated together or sequentially. Appendix I gives the timing protocol.

K Experimental protocol summary

Training and selection. For final evaluation, the proposed predictors, fixed-pole controls, and external baselines used a 200-epoch final-stage limit per seed across seeds 0–9, retaining the checkpoint with the lowest validation loss for each run. Hyperparameters, model variants, and component-count settings are optimized using validation data. Both chirp predictors use $K = 64$ on each forecasting dataset. The Gaussian-component models optimize the full-grid GP NLL: a denoising conditional for CMD-D and a mixture likelihood for the direct heads. Chirp-MSE diffusion and the mean-synthesizer comparison minimize MSE objectives. Representation pretraining and CMD-D's MSE warm start precede this final stage (Appendix C).

External-baseline hyperparameter selection. We conducted a grid search over 8–12 configurations per baseline, focusing primarily on the learning rate. Each configuration was optimized once with seed 0 for up to 100 epochs, enforcing early-stopping patience 10 and retaining its best-validation-loss checkpoint. We selected the configuration with the lowest validation loss under that model's native objective; losses were assessed only within the same architecture. These shorter trials selected hyperparameters and did not yield the reported test results. We deployed AdamW with gradient clipping at 1.0 and weight decay 0, except 10^{-3} for ProFITi.

Parameter-count scope. Table 18 records counts for this work's dataset-specific configurations selected through validation/grid search. The table is selective: it covers CMD, LLapDiff, and the listed external architectures; Table 19 details the direct structural controls. CMD counts include the trainable forecasting module and exclude the frozen VAE and history summarizer. CMD-X comprises 250,912 parameters (251,488 on BMS-air); CMD-D has 246,273 (249,393 on BMS-air), 0.83–1.85% fewer. External counts cover the reported selected model configurations.

External-baseline final training and testing. After hyperparameter selection, each baseline was retrained from scratch for the ten final seeds and evaluated at its best-validation-loss checkpoint. The 100-epoch limit and patience 10 applied only to the seed-0 search trials; final runs used the 200-epoch limit above with model-specific validation objectives.

Evaluation procedures. Appendices C–D specify forecasting, component pooling, the 25-path scoring procedure, and seed summaries. Appendix H details the separate synthetic training, validation, and evaluation streams and its learned and same-checkpoint controls.

Table 18: Trainable parameter counts for selected dataset-specific validation-search configurations, with scope defined above.

Model	USHCN	NOAA-UK	BMS-air	PhysioNet
CMD-D	246,273	246,273	249,393	246,273
CMD-X	250,912	250,912	251,488	250,912
LLapDiff	246,101	246,101	249,325	246,101
QuITE++ + DG	647,682	647,682	648,578	164,738
SDE-VI	364,138	364,138	365,496	309,234
TFMixer + DG	242,538	242,538	243,973	242,272
MOSES	290,576	290,512	290,512	290,512
ProFITi	249,479	283,019	283,019	249,223
CRU	80,277	90,973	87,381	83,101
NCDSSM	151,386	146,258	146,258	146,258
PatchTST + Gaussian	419,864	758,864	758,864	413,208

L SEPARATING FREQUENCY DRIFT, DECAY DRIFT, AND BASIS WIDTH

Experimental design. We evaluated which pole field must vary within the forecast window and how much harmonic flexibility the direct head requires. We used CMD-X on NOAA-UK ($h = 168$) and USHCN ($h = 12$), with $K = 64$ modes. For each dataset, we evaluated three drift settings—both fields varying, frequency only, and decay only—at basis widths $M_\phi \in \{2, 4, 8\}$. We also evaluated fixed-pole mixtures at $K = 64$ and $K = 79$, giving eleven configurations per dataset. We summarize test ES and VS by their mean and standard deviation across ten seeds per configuration. Fixed-pole results are reported in Table 1 and the full sweeps in Appendix E.

Basis width. The width counts signed harmonic features, rather than distinct frequencies. Width two uses the frequency $1/2$ with both signs; width four uses $\{1/2, 1\}$ with both signs; and width eight uses $\{1/2, 1, 3/2, 2\}$ with both signs. Frequencies are expressed relative to the harmonic time scale L. Thus width two permits a single half-cosine drift shape with either sign, and width four is the setting used in the main comparisons. Increasing the width expands the within-window pole trajectories available to the head.

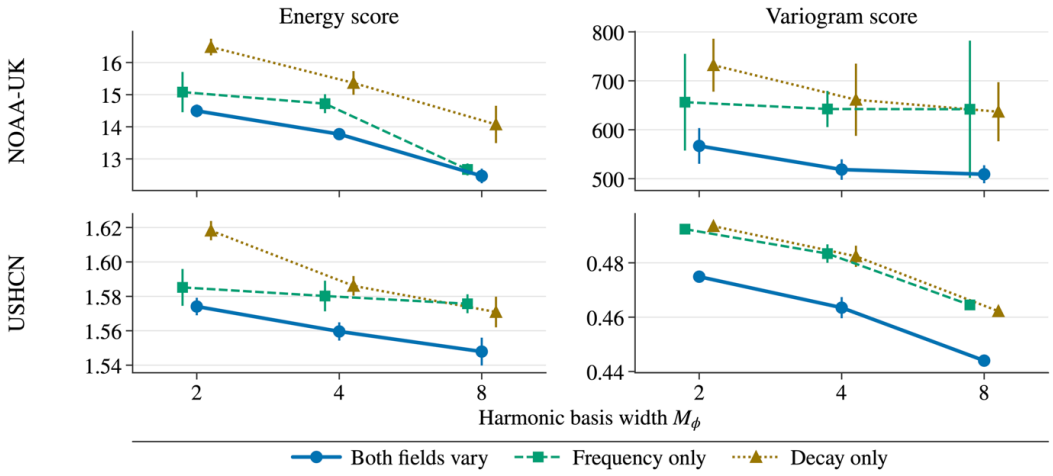


Figure 7: **Pole drift and harmonic basis width.** Mean ES and VS with one standard deviation over ten training seeds for CMD-X with 64 modes. Varying both frequency and decay gives lower mean joint scores than varying either field alone at every tested width. Fixed-pole comparisons are reported in Table 1 and Appendix E.

Table 19 records the counts associated with each setting. Both-field width four and single-field width eight have the same reported count of 250,912; the $K = 79$ fixed-pole head is within 0.04% at 250,826. The score comparisons below also show the effect of reallocating harmonic features between fields.

Table 19: **Reported parameter counts in the pole-drift ablation.** Counts are the same on NOAA-UK and USHCN; the forecasting-module scope is given in Appendix K. The frequency-only and decay-only settings have the same count.

Basis width	Both fields vary	One field varies
2	232,224	222,880
4	250,912	232,224
8	288,288	250,912
Fixed poles, $K = 64$	213,536	
Fixed poles, $K = 79$	250,826	

What we measured. Figure 7 shows that varying both fields gives lower mean ES and VS than varying either field alone at every tested width on both datasets. Increasing width from four to eight improves both-field ES by 9.45% on NOAA-UK and 0.75% on USHCN, and VS by 1.85% and 4.21%, respectively. Frequency-only drift gives lower ES than decay-only drift at all three widths on NOAA-UK and at widths two and four on USHCN; the ordering reverses at width eight on USHCN. The two pole fields therefore make different contributions, and their relative benefit depends on the dataset and metric.

Reallocating harmonic features between fields changes individual scores. On NOAA-UK, both-field width four has better VS than either single-field width-eight head, but frequency-only width eight has lower ES. On USHCN, both-field width four has lower ES than either single-field width-eight head, while decay-only width eight has slightly lower VS. These comparisons support retaining both fields as a useful default, while showing that reallocating capacity to a wider single-field basis can improve an individual score.

Additional harmonic flexibility. The wider direct head changes the ordering of some headline scores: both-field width-eight CMD-X has lower NOAA-UK ES than the width-four CMD-D configuration (12.469 versus 13.608), and lower USHCN VS (0.4440 versus 0.4541). It has higher NOAA-UK VS and higher USHCN ES. Added harmonic flexibility thus improves direct forecasts and changes which generator configuration attains the lowest score.

M LIMITATIONS AND FUTURE WORK

Our components assume independent latent coordinates and isotropic modal noise; cross-channel dependence emerges through the mixture and decoder. Dense likelihood evaluation scales cubically with the number of supervised queries per latent coordinate and component. The numerical covariance construction is valid on each requested grid, but consistency across query sets necessitates additional conditions (Appendix B.1). Future directions target richer latent dependence, efficient structured likelihood computation, and evaluation across additional domains and sampling patterns.